



University of Maribor
Faculty of Electrical Engineering and Computer Science
Smetanova ulica 17, 2000 Maribor
Slovenia
INSTITUTE OF ROBOTICS

DSP-2 Library for Simulink

User's Manual

Author:	Darko HERCOG
dsp2lib version:	1.5.1
Terminal version:	1.5.6
MATLAB version:	6.5
Contact:	darko.hercog@uni-mb.si

TABLE OF CONTENT

1.	PREFACE.....	1
2.	DSP-2 CONTROLLER	2
3.	DSP-2 LIBRARY FOR SIMULINK.....	5
3.1.	DSP-2 LIBRARY SETUP.....	6
3.2.	CREATING A NEW MODEL.....	7
3.3.	CODE GENERATION	8
3.4.	ONLINE CHANGING OF BLOCK PARAMETERS	11
3.5.	DSP-2 OPTIONS.....	15
3.6.	DSP-2 DEMOS	20
4.	DSP-2 BLOCK REFERENCES.....	22
4.1.	DSP-2 DEVICE DRIVER BLOCKSET	22
	<i>Analog Input.....</i>	<i>24</i>
	<i>Analog Input Oversampling.....</i>	<i>25</i>
	<i>Analog Output.....</i>	<i>27</i>
	<i>Analog Output Differential</i>	<i>28</i>
	<i>Digital input.....</i>	<i>29</i>
	<i>Digital Output</i>	<i>30</i>
	<i>From Address</i>	<i>31</i>
	<i>To Address</i>	<i>32</i>
	<i>PRBS</i>	<i>33</i>
	<i>Incremental Encoder.....</i>	<i>34</i>
	<i>Modulator.....</i>	<i>36</i>
	<i>PWM.....</i>	<i>37</i>
	<i>To File</i>	<i>38</i>
	<i>To Terminal.....</i>	<i>40</i>
	<i>From Terminal</i>	<i>42</i>
	<i>Transformations.....</i>	<i>44</i>
4.2.	DSP-2 ROBOTIC CONTROLLER DEVICE DRIVER BLOCKSET	45
	<i>Analog Input.....</i>	<i>46</i>
	<i>Analog Output.....</i>	<i>47</i>
	<i>Digital Input</i>	<i>48</i>
	<i>Digital Output</i>	<i>49</i>
	<i>Encoder</i>	<i>50</i>
	<i>Encoder Position Preset</i>	<i>52</i>
4.3.	DSP-2 PLANTS	54
5.	BLOCKS THAT ARE NOT SUPPORTED WITH <i>EMBEDDED C</i> FORMAT	55

TABLE OF FIGURES

Figure 1: Photograph of the DSP-2 controller	3
Figure 2: Block scheme of the DSP-2 controller	4
Figure 3: DSP-2 library for Simulink.....	5
Figure 4: Simulink Library Browser	6
Figure 5: new model GUI.....	7
Figure 6: New Simulink model.....	8
Figure 7: RTW page in MATLAB 6.5.....	9
Figure 8: Solver page	9
Figure 9: MATLAB command window - Successful completion of RTW build procedure	10
Figure 10: DSP Terminal – Options page.....	11
Figure 11: Tunable Parameters page in MATLAB 6.5.....	12
Figure 12: DSP Terminal – Parameters page	13
Figure 13: Simple Simulink model	14
Figure 14: GUI of Discrete-Time Integrator	14
Figure 15: DSP Terminal Parameters page	15
Figure 16: DSP-2 options (1)	16
Figure 17: DSP-2 options (2)	17
Figure 18: DSP-2 options (3)	17
Figure 19: DSP-2 options (4)	18
Figure 20: DSP-2 robotic controller Options (0)	18
Figure 21: DSP-2 robotic controller Options (1)	19
Figure 22: DSP-2 robotic controller Options (2)	19
Figure 23: DSP-2 robotic controller Options (3)	20
Figure 24: DSP-2 demos	21
Figure 25: DSP-2 device driver blockset	23
Figure 26: GUI of DSP-2 Analog Input block.....	24
Figure 27: GUI of DSP-2 AI Oversampling block.....	25
Figure 28: Automatically added objects to the Terminal GUI	26
Figure 29: GUI of DSP-2 Analog Output block.....	27
Figure 30: GUI of DSP-2 Digital Input block	29
Figure 31: GUI of DSP-2 Digital Output block	30
Figure 32: GUI of DSP-2 From Address block	31
Figure 33: GUI of DSP-2 To Address block	32
Figure 34: GUI of DSP-2 PRBS block	33
Figure 35: GUI of DSP-2 Encoder block	34
Figure 36: GUI of DSP-2 Modulator block.....	36
Figure 37: GUI of DSP-2 PWM block.....	37
Figure 38: GUI of DSP-2 To File block	38
Figure 39: GUI of DSP-2 To Terminal block	40
Figure 40: GUI of DSP-2 From Terminal block	42
Figure 41: DSP-2 robotic controller blockset.....	45
Figure 42: GUI of DSP-2 robotic controller Analog Input block	46
Figure 43: GUI of DSP-2 robotic controller Analog Output block	47
Figure 44: GUI of DSP-2 robotic controller Digital Input block.....	48
Figure 45: GUI of DSP-2 robotic controller Digital Output block.....	49
Figure 46: GUI of DSP-2 robotic controller Encoder block	50
Figure 47: GUI of DSP-2 robotic controller Encoder Preset block	52
Figure 48: ADDA interface card	54
Figure 49: DC motor with 3 phase H bridge.....	54

USED SYMBOLS

A/D	Analog to Digital converter
D/A	Digital to Analog converter
DSP	Digital Signal Processor
FPGA	Field Programmable Gate Array
GUI	Graphical User Interface
ISR	Interrupt Service Routine
PC	Personal Computer
PWM	Pulse Width Modulation
RTW	Real-Time Workshop
TI	Texas Instruments

GUI's

Name	Path
<i>Advanced page</i>	Simulink model > Simulation > Parameters > Advanced
<i>Diagnostic page</i>	Simulink model > Simulation > Parameters > Diagnostic
<i>Options page</i>	DSP Terminal > Options
<i>Parameters page</i>	DSP Terminal > Parameters
<i>RTW options page</i>	<i>RTW page</i> > category: DSP-2 options (i); i = 1..4
<i>RTW page</i>	Simulink model > Simulation > Parameters > Real-Time Workshop
<i>Solver page</i>	Simulink model > Simulation > Parameters > Solver
<i>Text page</i>	DSP Terminal > Text
<i>Tunable parameters page</i>	<i>Advanced page</i> > Configure... button
<i>Visual page</i>	DSP Terminal > Visual
<i>Robotic controller Options page</i>	<i>RTW page</i> > category: DSP-2 Robotic Options (i); i = 0..3

1. Preface

The present user's manual describes the *DSP-2 library for Simulink*. *DSP-2 library for Simulink* enables block programming of DSP-2 controller with simulation program MATLAB/Simulink®. The DSP-2 controller was developed at [Institute of Robotics](#) on [Faculty of Electrical Engineering and Computer Science](#), University of Maribor, Slovenia. The DSP-2 controller is a high performance, floating point digital signal processor (DSP) based inverter controller, designed primarily to control a three-phase AC motor. The controller is based on Texas Instruments TMS320C32 DSP and Field Programmable Gate Array (FPGA) XCS40-4PQ240C member of Xilinx Spartan Family (more information about DSP-2 controller can be found in DSP-2 user's manual).

Quick overview of each chapter in this user's manual:

Chapter 2 briefly describes main features of DSP-2 controller.

Chapter 3 describes *DSP-2 library for Simulink*. It is explained how to install this library to MATLAB environment, how to generate code from Simulink block diagram and how to change variables or parameters of Simulink model in online mode.

Chapter 4 contains full description of each block of DSP-2 device driver blockset and DSP-2 robotic controller blockset.

Chapter 5 contains a list of Simulink blocks that are not supported with Real-Time workshop Embedded Coder.

2. DSP-2 controller

As mentioned before the DSP-2 controller was developed at [Institute of Robotics](#) on [Faculty of Electrical Engineering and Computer Science](#). The DSP-2 controller is a high performance, floating point digital signal processor (DSP) based inverter controller designed primarily to control a three-phase AC motor. The controller is based on Texas Instruments TMS320C32 DSP and Field Programmable Gate Array (FPGA) XCS40-4PQ240C member of Xilinx Spartan Family.

The key features of DSP-2 controller are:

- DSP TMS320C32-60MHz
 - DSP serial interface
 - Two timer general purpose I/O pins
 - DSP MPSD interface for XDS510 emulator
- FLASH 256K x 8 - 70ns
- SRAM 128K x 32 0WS
- channel simultaneous 12bit A/D with serial output
 - Conversion and transfer to register in FPGA 2,6 μ s for all four channels
 - One channel with unipolar input range 0 to 4.095 V or 0 to 40,95 mA with 100 OHM shunt resistor
 - Two channels with bipolar input range -2.048 V to 2.047 V or -20,48 mA to 20.47 mA with 100 OHM shunt resistor
 - One channel with input multiplexer to select one of eight voltage input signals
 - First order input RC filters (time constant 33 μ s for i1 and i3 and 100 μ s for udc)
- Two channel 12bit D/A converter with serial input and unipolar output 0 to 4 V
- RS232 full duplex interface with fixed Baud Rate (57600 b/s, 8bits, 1stop, No parity)
- RS485 interface (not implemented in this firmware version (April 17, 2000))
- RS422 receiver for incremental encoder
- Three logic inputs and logic one output – all optically isolated (12 V passive)
- Bridge protection circuit
 - Interlock between bottom and top IGBT activation and dead time
 - Minimum pulse width, minimum pause width
 - In the presence of the fault signal the bridge is shut down unconditionally
- Three phase synchronous pulse width modulator
 - Twelve bit up/down counter for triangle generation
 - Symmetrical output pulses

- 66.6 ns time resolution
- generates interrupt pulses ones or twice in one modulator period
- Incremental encoder speed measurement with improved MT method
 - 66.6 ns time resolution
 - Position register contain position increment during sampling period
 - Time register contain relative time (in one modulator period) of last position change
 - Booth registers are saved on interrupt and are available until next interrupt
- Stand alone operation. Program is preloaded in FLASH.
- Operation with personal computer
 - Code Composer software development environment
- Standard RS232 serial interface (57600 Bd) and Terminal software
- MPSD interface for XDS510 emulator
- Multipoint communications RS485 and CAN
- Controller dimensions 161 x 130 mm

More about DSP-2 controller can be found in DSP-2 user's manual.

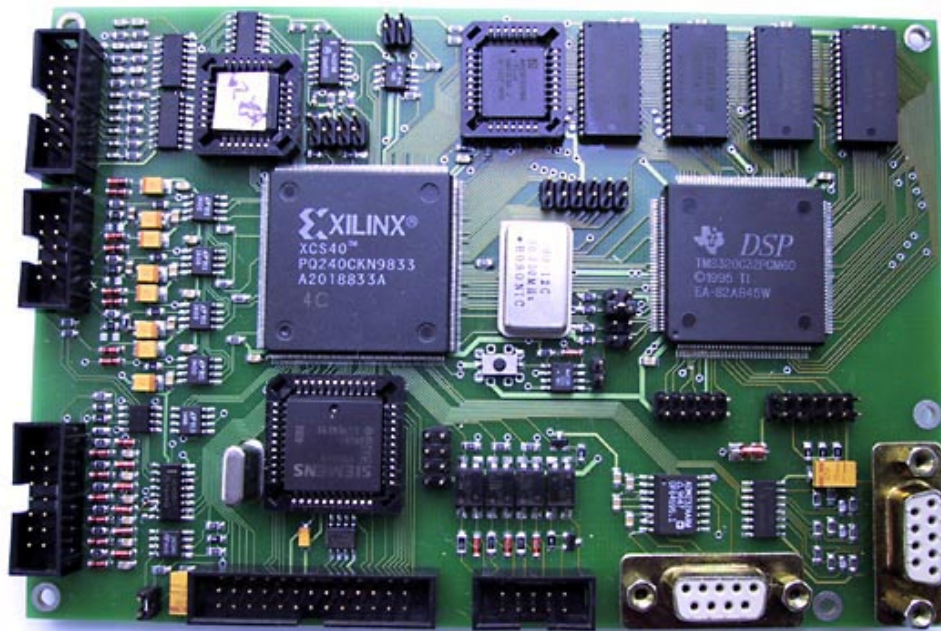


Figure 1: Photograph of the DSP-2 controller

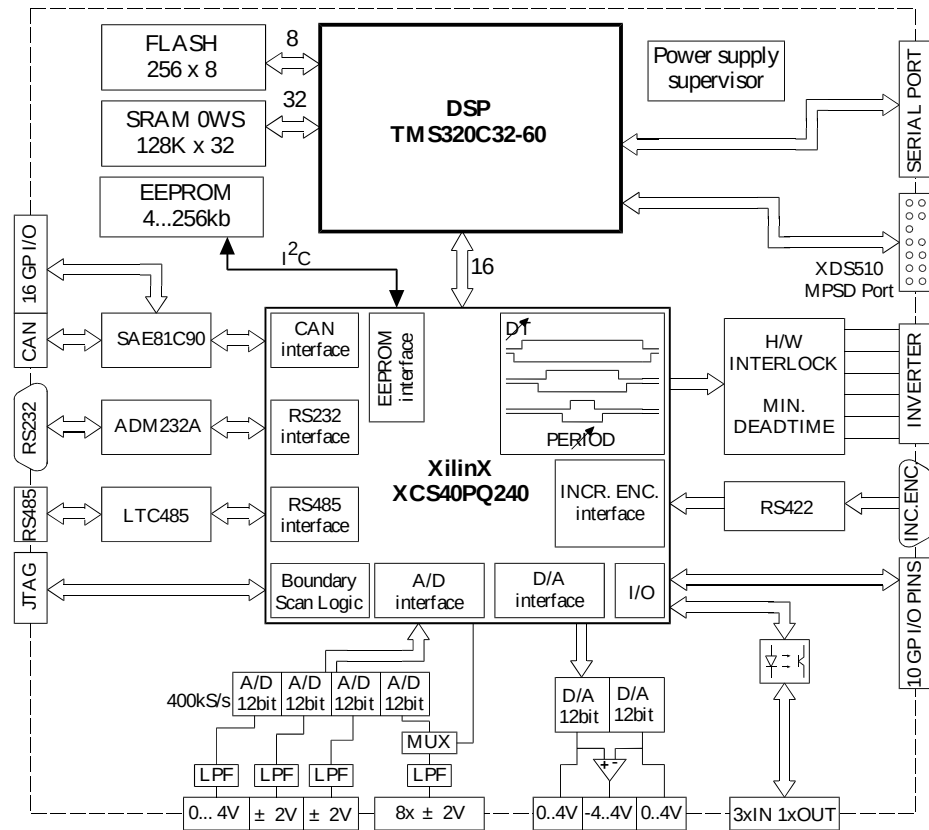


Figure 2: Block scheme of the DSP-2 controller

3. DSP-2 library for Simulink

After successful installation of the DSP-2 library (installation process is explained in chapter 3.1), this library can be opened with **dsp2lib** command in MATLAB command window. DSP-2 library will appear (Figure 3). It contains the following blocks:

- *Basic Blockset*: subsystem contains the DSP-2 device driver blockset (Figure 25);
- *Robotic Blockset*: Subsystem contains Simulink blocks for DSP-2 robotic controller (Figure 41);
- *Doc*: DSP-2 documentations in pdf format;
- *Demos*: link to DSP-2 basic examples;
- *New model*: more about this option can be found in chapter 3.2 on page 7;
- *About*: general information's about *DSP-2 library for Simulink*;

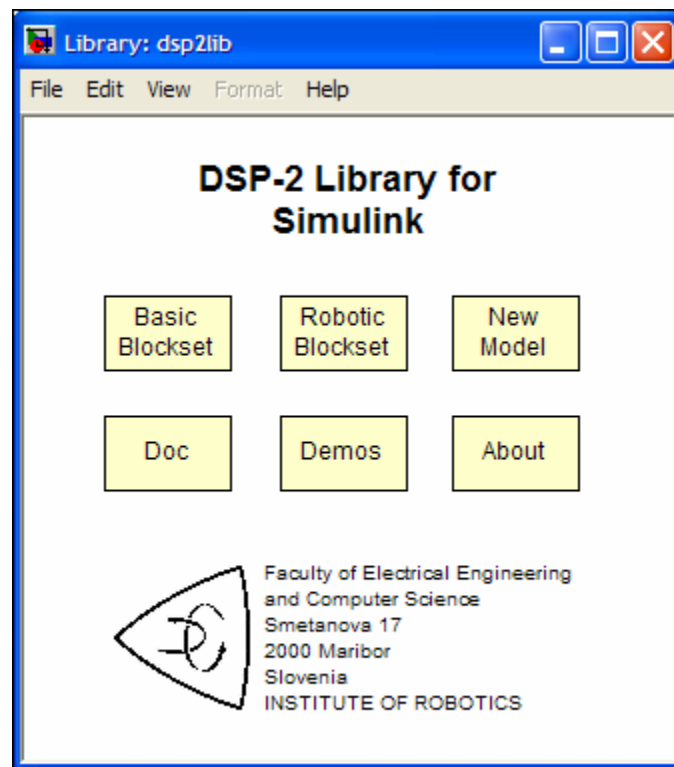


Figure 3: DSP-2 library for Simulink

Note: Blocks in *basic* and *robotic blockset* can also be accessed in Simulink Library Browser. Just open the Simulink and click on plus sign near *DSP-2 controller* text (Figure 4).

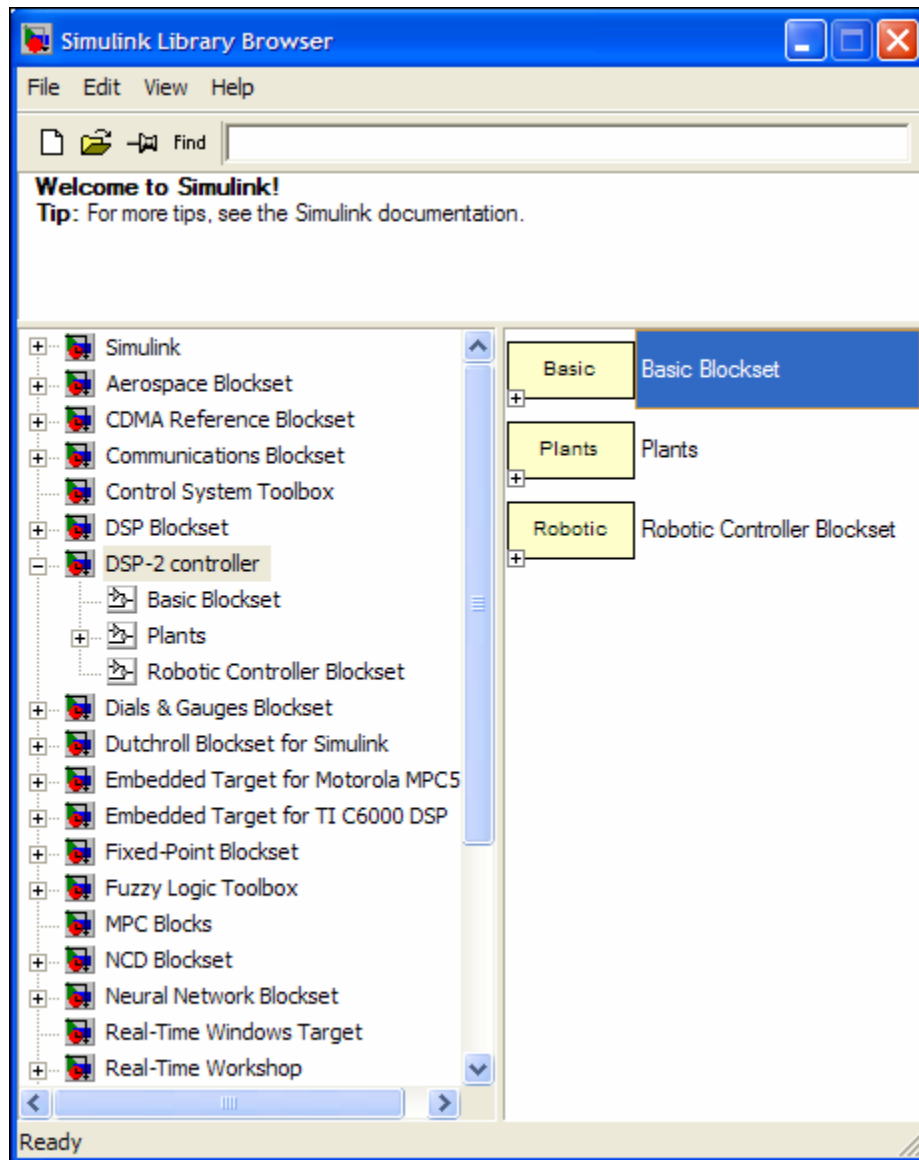


Figure 4: Simulink Library Browser

3.1. DSP-2 library setup

Before the installation of DSP-2 library for Simulink, the following software must have been installed on personal computer:

- MATLAB® 6.5, 6.5.1 or 6.5.2
- Real-Time Workshop®
- Real-Time Embedded Coder®
- TI compiler for C3x4x family of DSP processors (Code Composer)

Installation process

Put CD with DSP-2 library in your CD drive. After a while setup GUI will appear. It is recommended, that you select default settings in all GUI's that will appear in setup process. Setup program will copy all program files and documentations to *C:\FERI\dsp2lib\matlab6p5* folder. The shortcut to DSP Terminal will be created on desktop.

Deinstallation process

In Control panel select *Add or Remove Programs*. In *Add or Remove Programs* window find *DSP-2 Library for Simulink*, and select *Remove*.

3.2. Creating a new model

If you want to build a new Simulink model, which will target DSP-2 controller, it is recommended that you double click the block *New Model* in the DSP-2 library. After this GUI in which you must enter new model name will appear (Figure 5). All the model parameters will be initialized so that the *Build process* will successfully generate executable code suitable for DSP-2 controller from Simulink model.

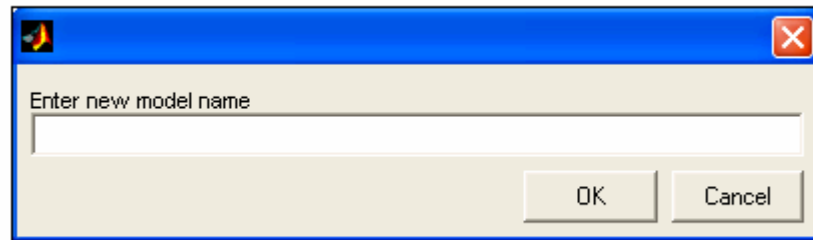


Figure 5: new model GUI

After entering model name, new Simulink model will appear (Figure 6). Now you can add DSP-2 blocks or built-in Simulink blocks in new Simulink model.

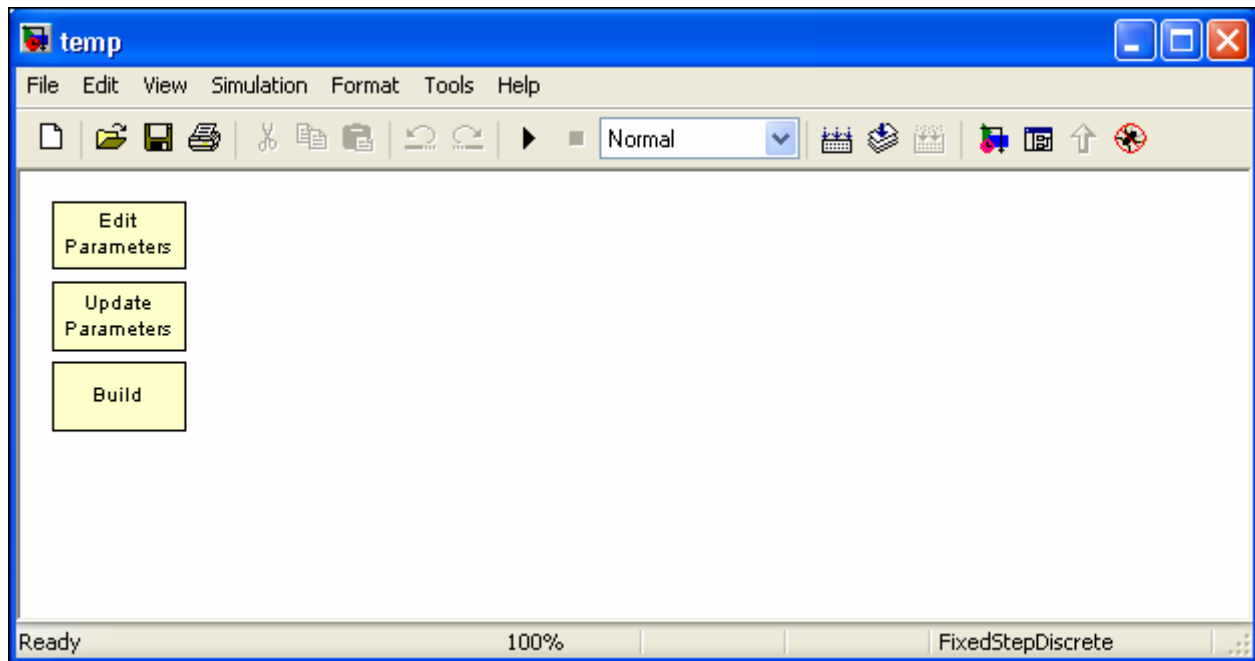


Figure 6: New Simulink model

3.3. Code generation

How to generate executable code from Simulink model? Just open *RTW page* (Figure 8). If you created a new model as described in chapter 3.2, then all parameters are predefined so, that you just click on *Build* button and code generation process will start. Otherwise you must set the following parameters:

Real-Time Workshop page:

On *RTW page* in *Category* option select *Target configuration*. The *Category* section will appear. Set parameters to the following values:

- o System target file: **ert_dsp2.tlc**
- o Template makefile: **ert_dsp2.tmf**
- o Make command: **make_rtw**

Solver page:

Simulation time:

- Start time: **0**
- Stop time: **inf**

Solver options:

- Type: **Fixed-step** and **discrete (no continuous states)**

- Fixed step size: this parameter define **interrupt period**. It can be set to 50e-6 or greater, but smaller than 250us.
- Mode: **auto**

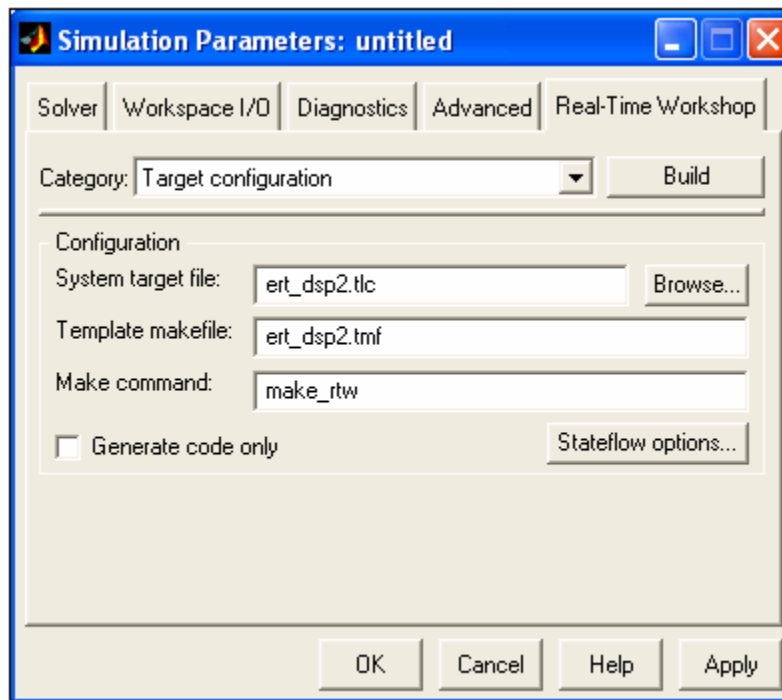


Figure 7: RTW page in MATLAB 6.5

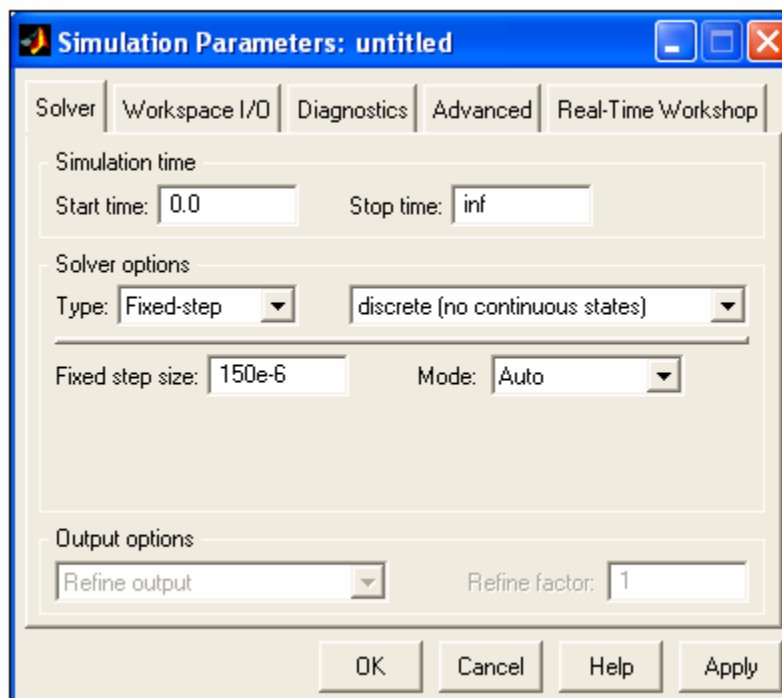


Figure 8: Solver page

When all the mentioned parameters are set to the correct values, you can start a build process with single click on *Build* button on *RTW page*. After successful compilation the following message in MATLAB command window will appear (Figure 9).

```
### Successful completion of the ERT code generation process for the DSP-2 board!
```

Figure 9: MATLAB command window - Successful completion of RTW build procedure

Now you must run the *DSP Terminal* (the shortcut is located on the desktop) to download generated code to DSP-2 controller. In *DSP Terminal Options page (1)* (Figure 10) you can navigate to directory in which the generated code is located. Note that generated code appears with extension *.hex*. Select appropriate file and press *Download* button to download code to DSP-2 controller. If the download is not possible, first reset the DSP-2 controller with *Reset* button.

If an option *Download generated code to the DSP-2 controller* on the *RTW options page (1)* (Figure 16) is selected, then downloading process is automatically invoked.

```
### Successful completion of the ERT code generation process for the DSP-2 board!  
  
### Downloading generated code to the DSP2 board...
```

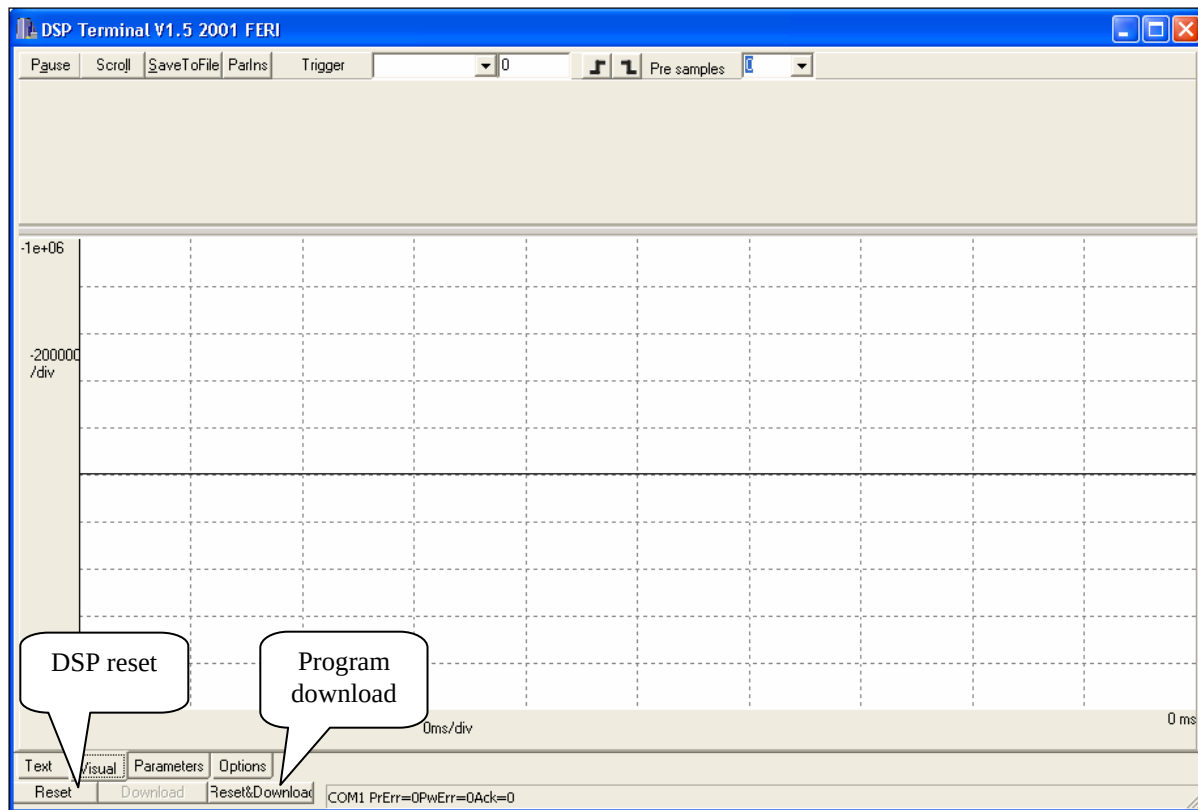


Figure 10: DSP Terminal – Options page

3.4. Online changing of block parameters

DSP-2 library in combination with DSP Terminal enables online changing of the Simulink block parameters, while the code is executed on the DSP-2 controller. Parameters of blocks, which are placed in Simulink model, must be stored in local or global variable in MATLAB workspace as **real scalar** values.

Which parameters are changeable in online mode depends on *Inline parameters* button on *Advanced* page:

- If *Inline Parameters* button on the *Advanced* page is NOT selected, than all scalar parameters of each block placed in the Simulink model will appear in the *Parameter Inspector* window of DSP Terminal (). When the Simulink model is pretty complex, a lot of parameters appear in the *Parameter Inspector* window. Usually we want to change only a few parameters in online mode, therefore option, described in the next section is much more appropriate.
- If the *Inline Parameters* button is selected, than you can define which parameters you want to tune in online mode. This can be done on *Tunable parameters* page (Figure 11). This page appears, if you select: *Configure...* button on *Advanced* Page. The list of MATLAB workspace variables is placed on the left side of the *Model Parameter Configuration* window (Figure 11). Select the variable which you want to change in online mode and click *Add to Table...* button to add parameter to *Global*

(tunable) parameters table. Parameters *Storage Class* and *Storage type qualifier* leave to the default values.

Because an example is the best way to understand the theory of operation, an example is presented in the subsection.

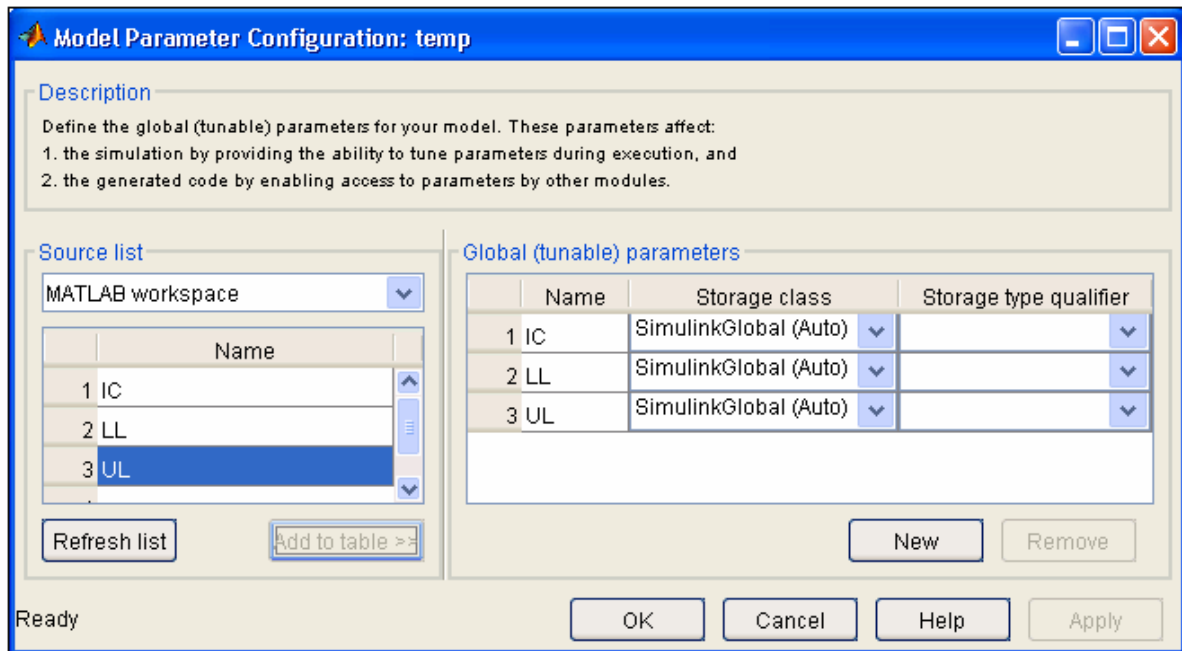


Figure 11: Tunable Parameters page in MATLAB 6.5

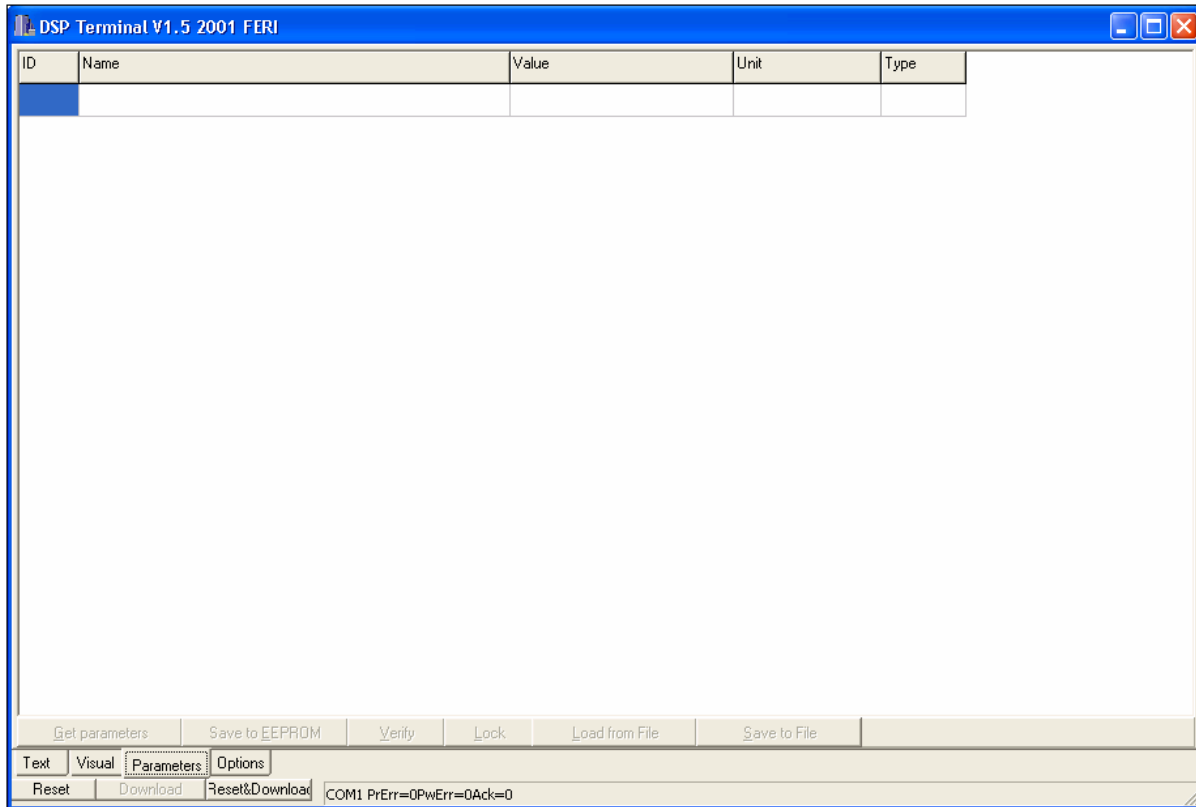


Figure 12: DSP Terminal – Parameters page

Example

In this example we'll show how to define exported parameters. We have a Simulink model shown in Figure 13. In Discrete Time integrator we define parameters, as shown in Figure 14 (the parameters that we want to export are in GUI of Discrete Time Integrator signed as IC, LL, and UL). If we want to change these parameters in online mode, we must add these parameters to *Tunable parameters page* (Figure 11). Note that parameters UL, LL and IC must be defined in MATLAB workspace, before you start build procedure. In our example this parameters has the following values: IC=50; UL=100; LL=0;

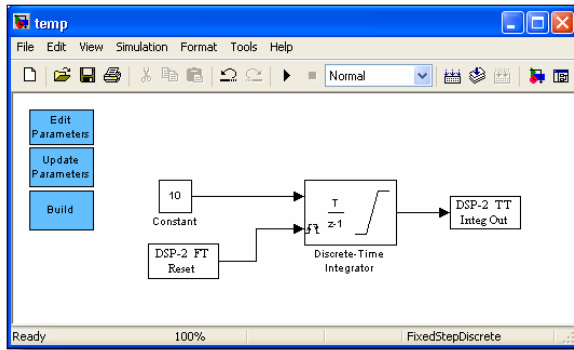


Figure 13: Simple Simulink model

Figure 14: GUI of Discrete-Time Integrator

Now we can build the code from Simulink model and download the generated code to DSP-2 controller. After a few seconds we can see, that in DSP Terminal *Parameters* page the same parameters, as we assigned in *Tunable parameters* page (Figure 15), appear. Now you can change parameters values (if this is not possible use *Unlock* button), by entering a different value to *Value* cell near parameter *Name* cell, while the generated code is executing on the DSP-2 controller. In DSP Terminal *visual* page you can observe effects of changes.

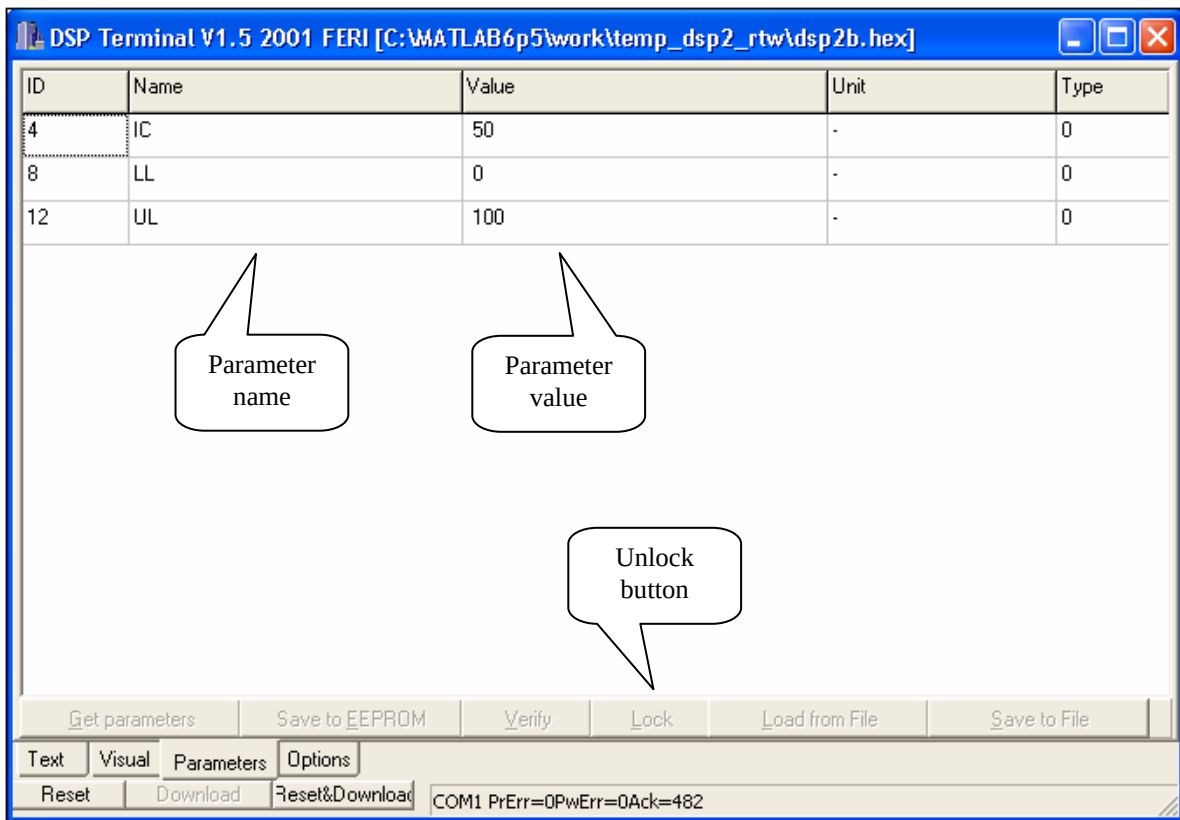


Figure 15: DSP Terminal Parameters page

3.5. DSP-2 options

In this section some additional DSP-2 options are explained. These options are placed on the *RTW options* page.

DSP-2 options (1) (Figure 16):

- *Download generated code to the DSP-2 controller*: if this option is selected, then DSP Terminal is automatically invoked when RTW terminates with the code generation process;
- *Create LabVIEW virtual instrument*: If LabVIEW 7.0+ and DSP-2 add-on toolkit for LabVIEW are installed on PC, LabVIEW virtual instrument will be automatically created during executable code generation. More information's about this option can be found in *LabVIEW virtual instrument for DSP-2 controller* document.
- All other options on this page are Simulink built-in options, therefore explanation of these options can be found in the Simulink documentation.

DSP-2 options (2) (Figure 17)

Individual options of this page are explained in the description of the related block.

DSP-2 options (3) (Communication) (Figure 18)

Currently only one option exists on this page. Option *CAN Baud Rate* defines baud rate for CAN bus communication. This option must be initialized to the appropriate value, when the DSP-2 CAN Read and/or DSP-2 CAN Write blocks are used in the Simulink model.

DSP-2 options (4) (External mode) (Figure 19)

Options on this page are NOT implemented yet! Parameter *External mode* MUST be UNSELECTED!

DSP-2 robotic controller options (i) (i=0..3)

These options are explained in DSP-2 robotic controller device driver blockset section on page 45.

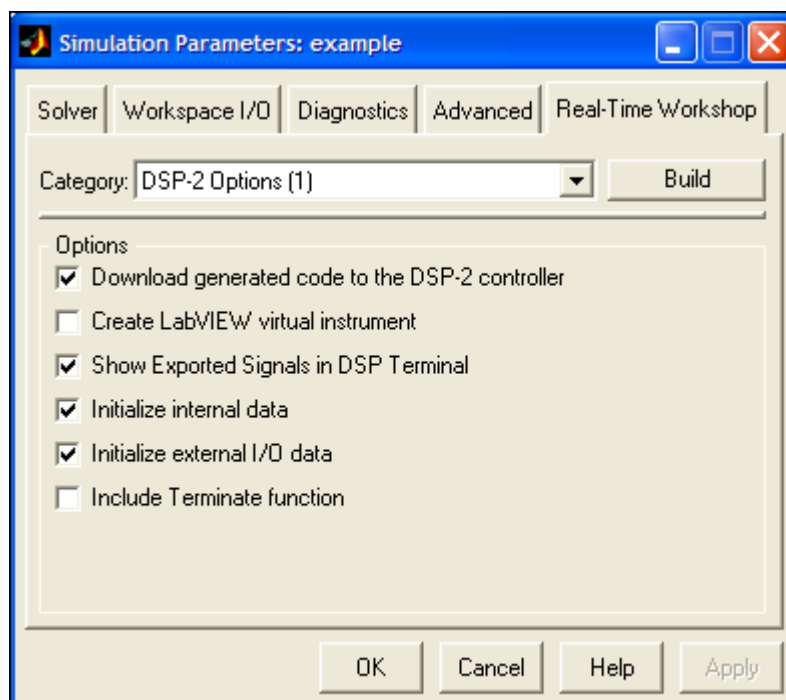


Figure 16: DSP-2 options (1)

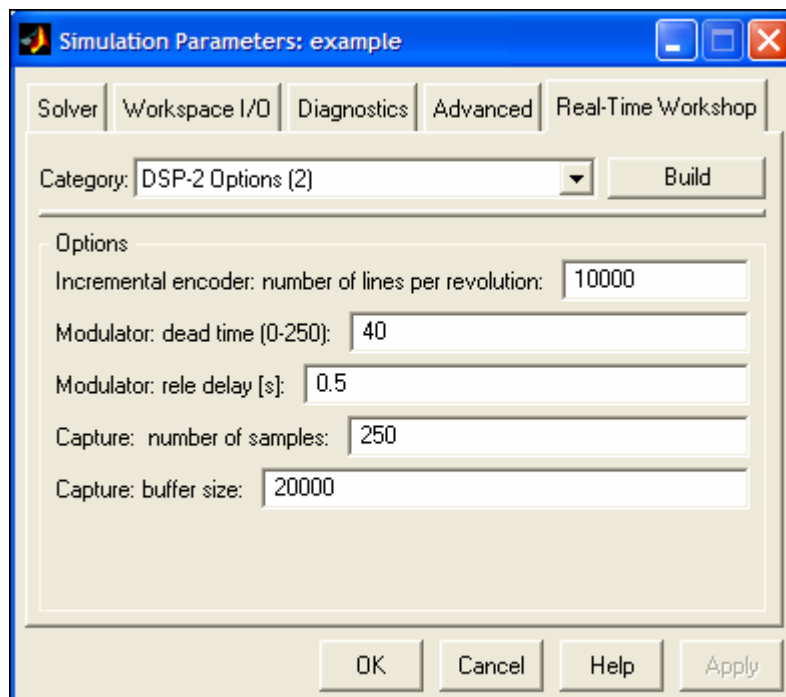


Figure 17: DSP-2 options (2)

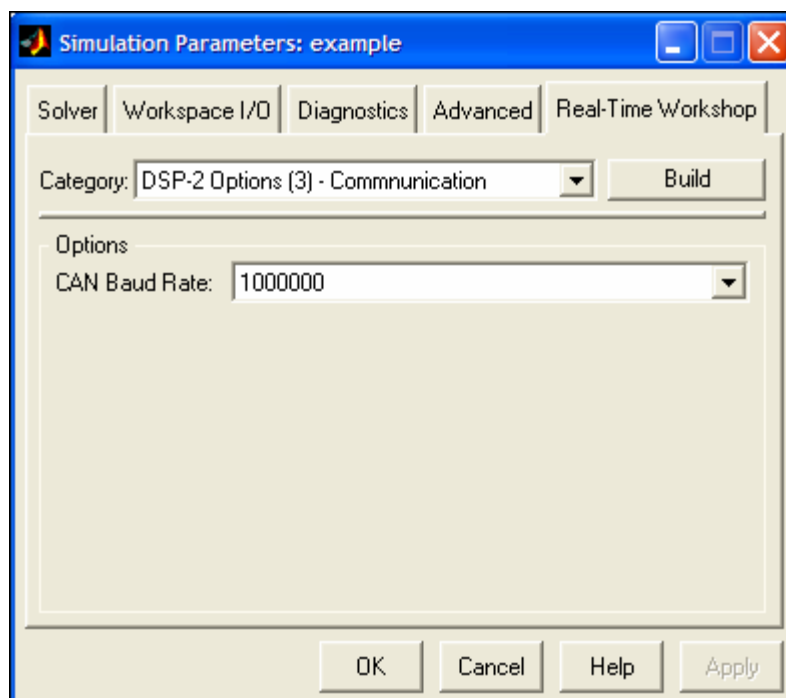


Figure 18: DSP-2 options (3)

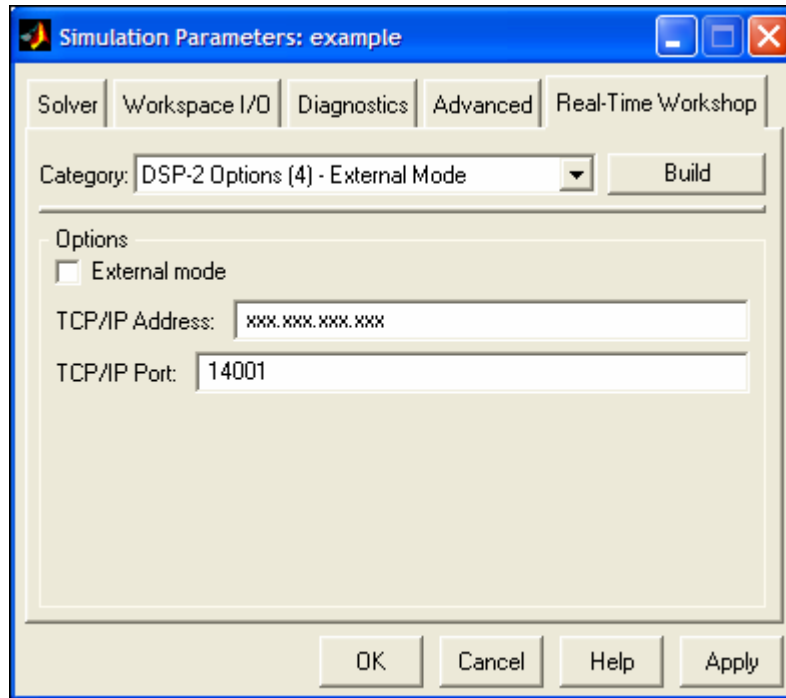


Figure 19: DSP-2 options (4)

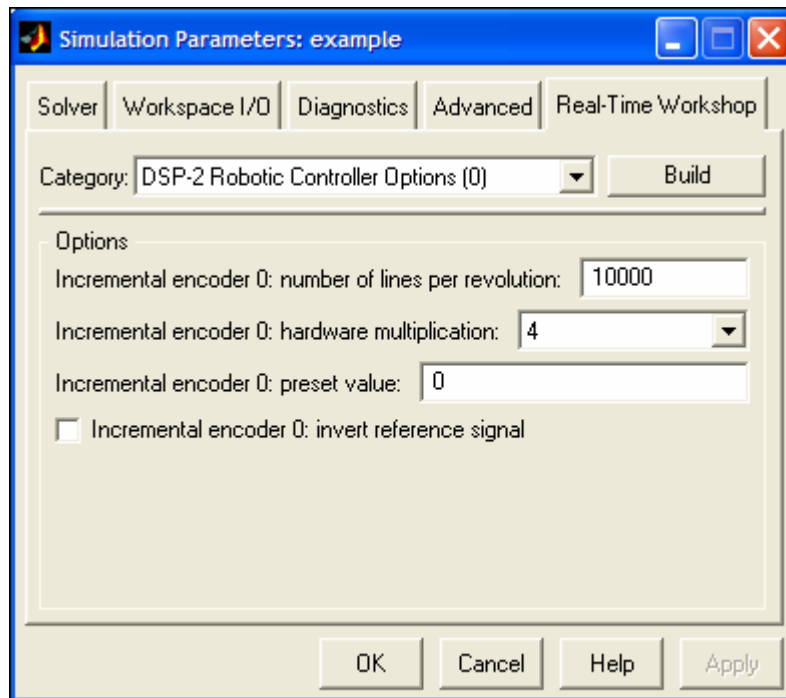
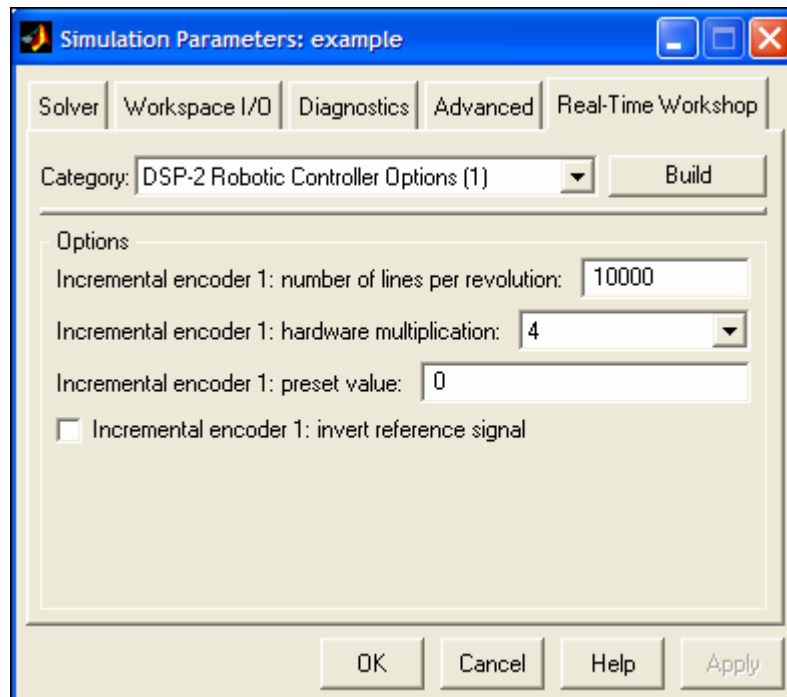
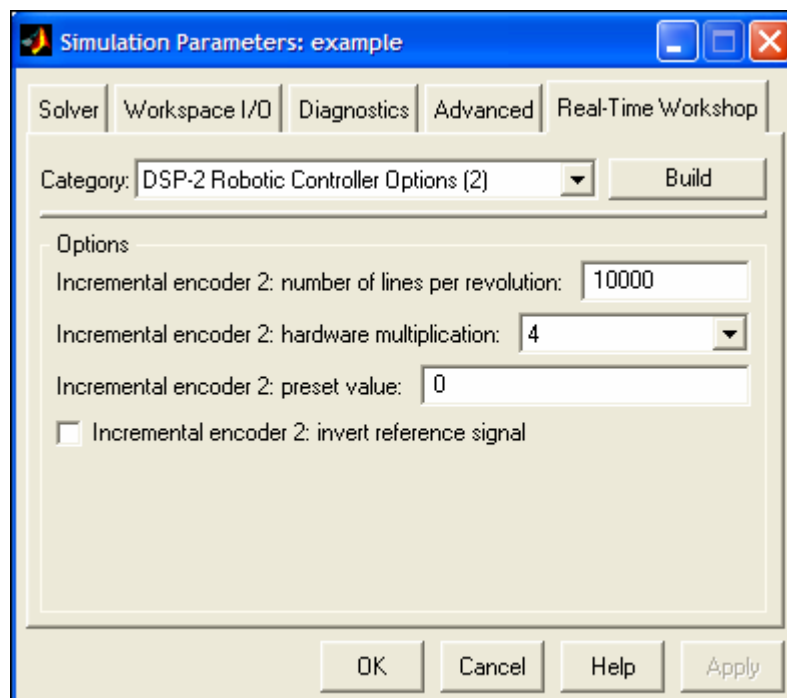


Figure 20: DSP-2 robotic controller Options (0)

*Figure 21: DSP-2 robotic controller Options (1)**Figure 22: DSP-2 robotic controller Options (2)*

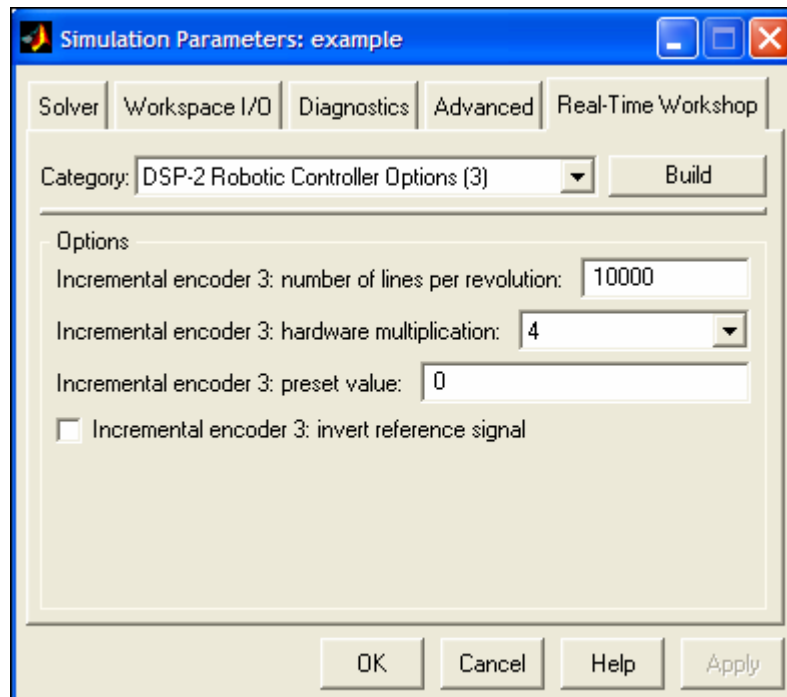


Figure 23: DSP-2 robotic controller Options (3)

3.6. DSP-2 demos

DSP-2 Library for Simulink contains a set of Simulink demo models. These demos are available at MATLAB demos in the Blocksets section (Figure 24).

To access DSP-2 demos, enter *demos* in MATLAB command window and then navigate to *Blocksets* -> *DSP-2 Library for Simulink* (Figure 24).

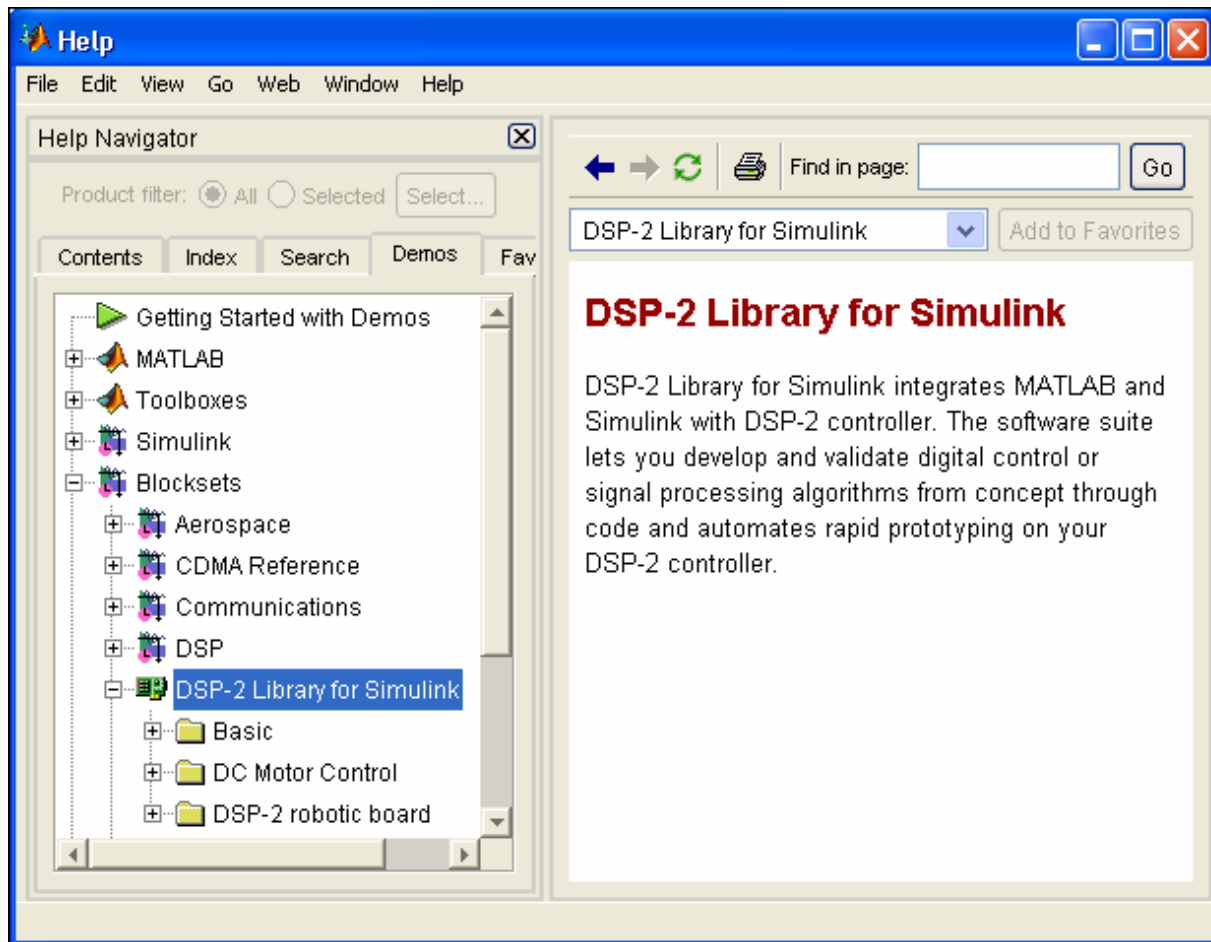
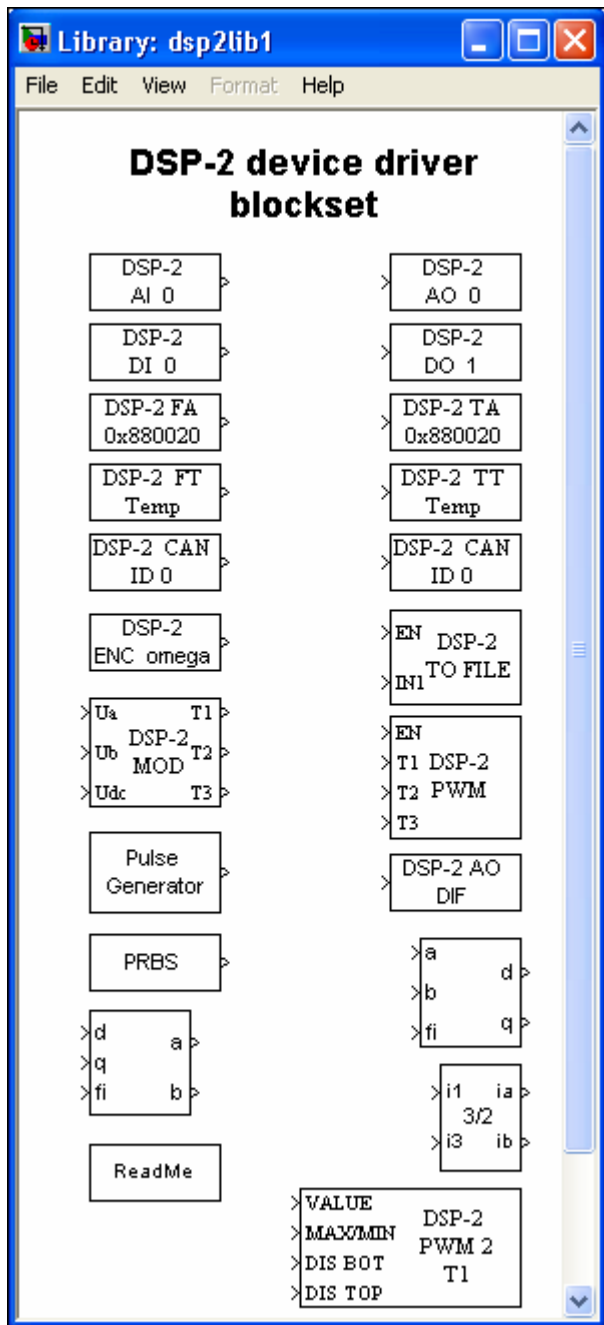


Figure 24: DSP-2 demos

4. DSP-2 block references

4.1. DSP-2 device driver blockset

In the following subsections detailed description of individual DSP-2 device driver blocks are explained.



DSP-2 device driver blockset contains the following blocks:

- Analog Input
- Analog Input Oversampling
- Analog Output
- Analog Output Differential
- Digital Input
- Digital Output
- From Address
- To Address
- PRBS
- Incremental Encoder
- Modulator
- PWM
- To File
- To Terminal
- From Terminal
- Transformation blocks

Figure 25: DSP-2 device driver blockset

Analog Input

Description: DSP-2 controller has three fast analog inputs (signed with 0, 1, 2 in GUI) and one “slow” analog input, which has an 8/1 multiplexer placed on its input (signed with 3[0] to 3[7] in GUI). Algorithm of this block is made so, that at the start of each ISR DSP performs sampling of all three fast analog inputs and one of eight signals that are connected to multiplexer. In each interrupt, sampling of different port of “slow” analog input is executed. Because multiplexer type is 8/1, that means, that at every 8-th interrupt sampling on the same port of »slow« analog input is performed. Voltage ranges of analog inputs are shown in table below (Table 1). All A/D converters are 12-bit, that means, that on output of A/D converter we get a unipolar input range signal from 0 to 4095 quants, while at bipolar from – 2048 to 2047 quants. On the output of this block we get value of quants stated in table below.

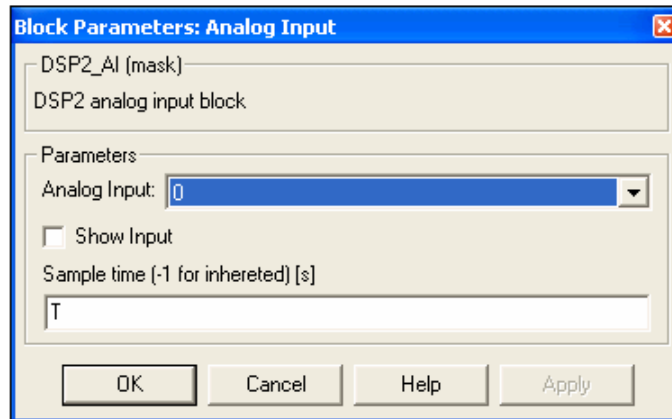
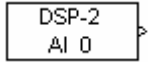


Figure 26: GUI of DSP-2 Analog Input block

Table 1: Analog input ranges

DSP-2 analog input	DSP-2 input range [V]	A/D converter output [quant]
0	0 to 4.095	0 to 4095
1	-2.048 to 2.047	-2048 to 2047
2	-2.048 to 2.047	-2048 to 2047
3[0] to 3[7]	-2.048 to 2.047	-2048 to 2047

Parameters: Analog input

Analog input, from which DSP reads data

Sample time

Sample time must be an integer multiple of the base sample time (*Fixed sample time* parameter).

Analog Input Oversampling

Description: Analog Input Oversampling block enables oversampling of the signals connected to the analog inputs 0, 1, 2 of DSP-2 controller. Oversampling means, that these signal are measured n times during one period of PWM signal.

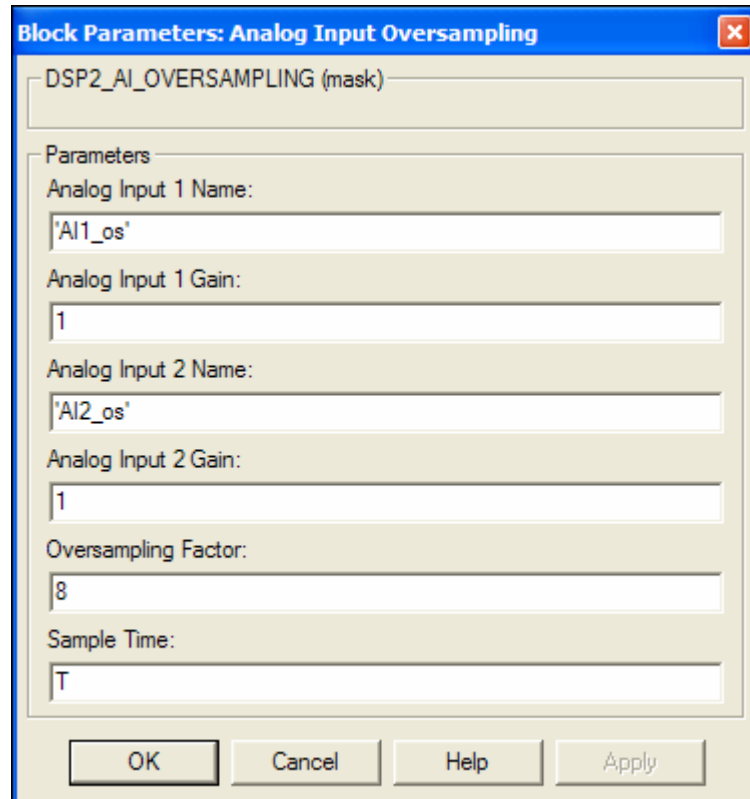
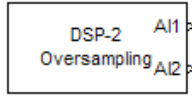


Figure 27: GUI of DSP-2 AI Oversampling block

Outputs: **Outputs AIX (X=1, 2)**

Output AIX (X=1,2) is oversampled value of the signal connected to Analog Input X. Oversampling means, that the signal is measured n times during one period of PWM signal. Output AIX (X=1, 2) is the vector with n elements. Dimension of this vector is defined with 'Oversampling Factor' parameter in block GUI.

Parameters: **Analog Input X Name (X=1, 2)**

For each output channel, the user can assign custom name. This name will appear in Terminal GUI after executable code deploying on the DSP-2 target.

Analog Input X Gain (X=1, 2)

Each sampled value in analog input channel X (X=1, 2) can be scaled with parameter Gain.

Oversampling Factor

Oversampling factor defines how many times the signal will be measured in one period of PWM signal.

Parameter range: $1 < \text{Oversampling Factor} < 8$

Default value: 1

Note: In the 'Oversampling Factor' field the user can enter the variable name. If the variable is not yet defined in MATLAB workspace, the size of each output vectors will be automatically set to 1.

Sample time

Sample time must be an integer multiple of the base sample time (*Fixed sample time* parameter).

Additional notes

After deploying executable code, generated from Simulink model which contains 'DSP-2 Analog Input Oversampling' block, one numerical control ('OS MODE') and two numerical indicators automatically appear in DSP Terminal GUI (Figure 28). The names of indicators are equal to the names entered in *Analog Input X Name* (X =1, 2) field in block GUI.

If the variable 'OS MODE' (Oversampling Mode) is set to 0 then all in graph presented variables are sampled on the DSP-2 controller with the 'basic sampling period'. This period is equal to 'Fixed step size' parameter in Simulink model.

If the variable 'OS MODE' (Oversampling Mode) is set to 1 then all in graph presented variables are sampled with the 'oversampling period'. This period is defined as:

$$\text{oversampling period} = \frac{\text{basic sampling period}}{\text{oversampling factor}}$$

Note: Note: If 'OS MODE' is set to 0, this does not mean, that the block output vectors are empty. Oversampling is executing independent from 'OS MODE' variable! If 'OS MODE' is set to 0, then all in graph presented variables are sampled with 'basic sampling period'

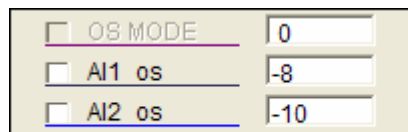
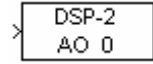


Figure 28: Automatically added objects to the Terminal GUI

Analog Output

Description: DSP-2 controller has only two analog outputs. With parameter *Analog Output* (Figure 29) can be selected, on which output DSP sends block will input data. Both D/A converters are 12-bit and have the output range from 0 to 4.095V. If block input signal is greater than 4095 quants, or smaller than 0, the signal is limited to 4095 or 0 respectively.



Caution: If at least one of the DSP-2 Analog Output block exists in the Simulink model, than block AO_DIFF is not allowed to be in the model.

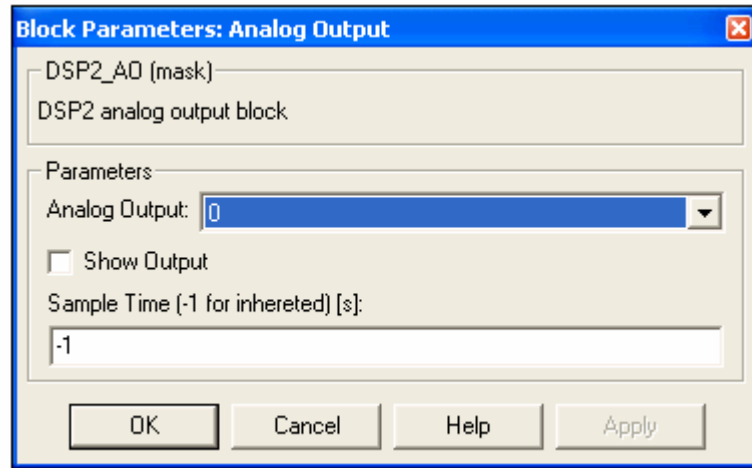


Figure 29: GUI of DSP-2 Analog Output block

Table 2: Analog output ranges

DSP-2 analog output	DSP-2 analog output range [quants]	D/A converter output [V]
0	0 to 4.095	0 to 4.095
1	0 to 4.095	0 to 4.095

Parameters: Analog output

We can select to which analog output on DSP-2 controller, DSP will send block input data

Show input:

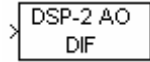
This option is used only in the simulation. If option is selected, than block input appears. In the simulation block input values are copied to the block output.

Sample time:

Sample time must be an integer multiply of the base sample time (*Fixed step size* parameter). If sample time is set to -1, then sample time is inherited from driving block. Default value is '-1'.

Analog Output Differential

Description: Block Analog Output Differential enables bipolar analog output generation on the *AOUT* port of the DSP-2 controller.



Caution: If this block exists in the Simulink model, than block Analog Output (DSP-2 AO) is not allowed to be in the model.

Table 3: Analog Output Differential output range

DSP-2 analog output	Block input range [quants]	DSP-2 output range [V]
AOUT	-4096 to 4.095	-8 to 8

Parameters: None.

Digital input

Description: Figure 30 shows GUI of DSP-2 Digital Input block. DSP-2 controller has 3 optically isolated logical inputs. Which input to use, can be selected with the parameter *Digital input*. On the output of this block appears the signal, which has the logical value '0' or '1'.

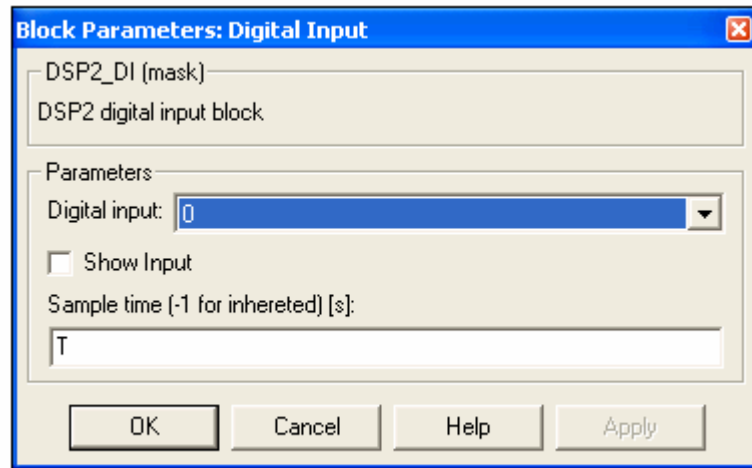
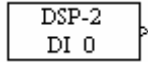


Figure 30: GUI of DSP-2 Digital Input block

Parameters: Digital input

Digital input of DSP-2 controller, from which DSP gets data

Show input

The meaning of this parameter is the same as in DSP- 2 Analog Input block

Sample time

Sample time must be an integer multiple of the base sample time (*Fixed sample time* parameter).

Digital Output

Description: Figure 31 shows the GUI of the digital output block. DSP-2 controller has only one optically isolated logic output. In the GUI of this block you can set the following parameters: *Digital output*, *Threshold* and *Operation*. The value of the output signal is depending of height of the input signal, operation and threshold.

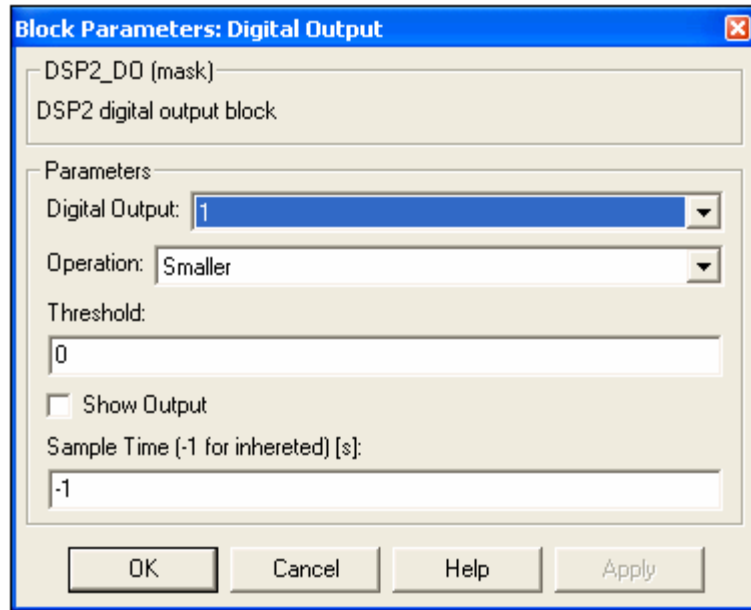
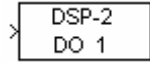


Figure 31: GUI of DSP-2 Digital Output block

Parameters: Digital output

Select to which digital output DSP will sends data

Operation

Greater : If the input signal is greater than the threshold, then the output is set to '1' else to '0'

Smaller: If the input signal is smaller than the threshold, then the output is set to '1' else to '0'

Threshold

Threshold must be a scalar real number. The meaning of this parameter is explained in *Operation* parameter.

Sample time

Sample time must be an integer multiply of the base sample time (*Fixed step size* parameter). If the sample time is set to -1, then the sample time is inherited from the driving block. Default value is -1.

From Address

Description: Block DSP-2 From Address (Figure 32) enables digital signal processor to get the data from the address in memory. In GUI of DSP-2 From Address block you can set two parameters: *Address* and *Initial value*. We can select *Address* only from drop down menu, because this disables input of the wrong address. Parameter *Initial value* defines the value, to which the selected address is set in the initialization stage of the DSP-2 controller. Default value of mentioned parameter is 0.

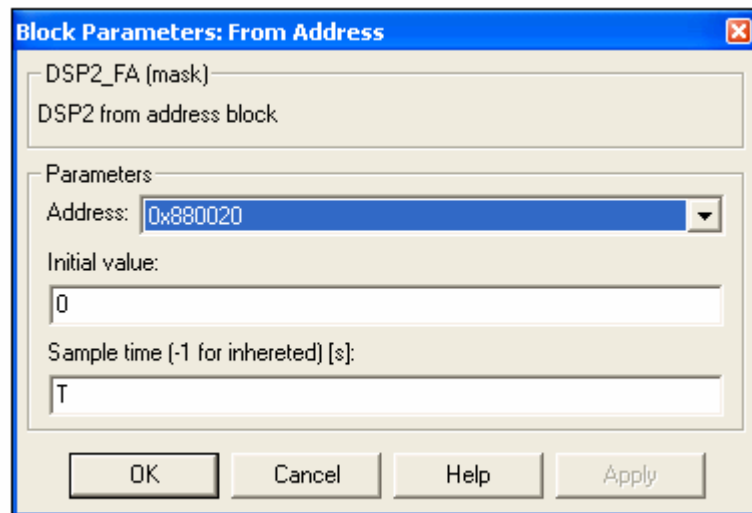
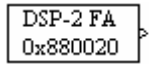


Figure 32: GUI of DSP-2 From Address block

Parameters: Address

Address in DSP-2 memory, from which DSP gets data.

Initial value

The value to which DSP initializes address (selected by *Address* parameter) in DSP-2 memory in the initialization stage of DSP-2 controller.

Sample time

Sample time must be an integer multiple of the base sample time (*Fixed sample time* parameter).

To Address

Description: This block is similar to DSP-2 From Address block with one exception. This block is sink block, and enables digital signal processor to store the block input value to the address specified in the parameter *Address* in "float" format (Figure 33).

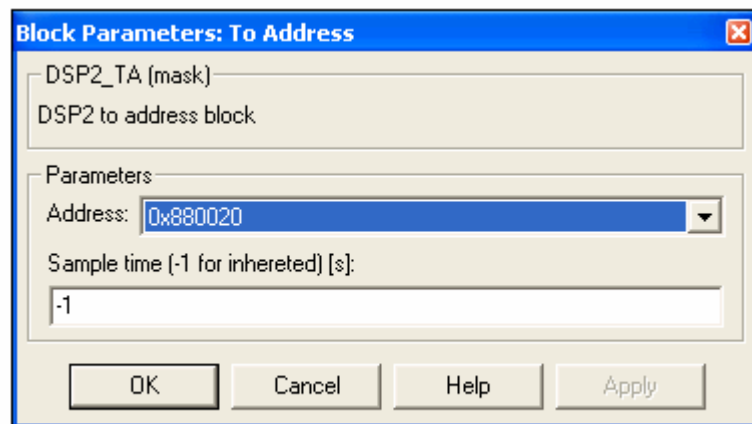
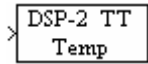


Figure 33: GUI of DSP-2 To Address block

Parameters: Address

Address in DSP-2 memory, to which DSP stores the block input data

Sample time

Sample time must be an integer multiply of the base sample time (*Fixed step size* parameter). If the sample time is set to -1, then the sample time is inherited from the driving block. Default value is '-1'.

PRBS

Description: PRBS block generates Pseudo Random Bipolar Sequence, which is mainly used in the system identification. In GUI you can set the following parameters: *Amplitude*, *Mean value* and *Sample time*.

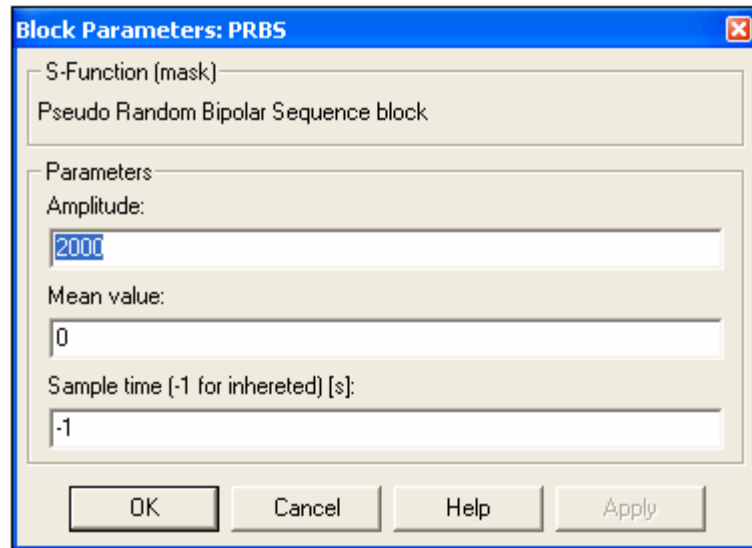


Figure 34: GUI of DSP-2 PRBS block

Parameters: Amplitude

Parameter *Amplitude* defines amplitude of generated prbs signal. Generated signal has two output states:

- $\text{Mean value} + \text{Amplitude}$
- $\text{Mean value} - \text{Amplitude}$

Mean value

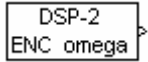
Mean value of the generated prbs signal

Sample time

Sample time must be an integer multiple of the base sample time (*Fixed sample time* parameter).

Incremental Encoder

Description: Block DSP-2_ENC enables sampling of the position and speed of an incremental encoder that is connected to DSP-2 controller. Figure 35 shows the GUI of DSP-2 Encoder block. In this GUI you can set two parameters: *Mode* and *Gain*. The meaning of these parameters is explained in Parameters section below.



To ensure the correct operation of incremental encoder, you must set parameter *Num of pulses per revolution*, which is located on *RTW Options page (2)* (Figure 17). This parameter appears only, if DSP-2 target is selected. As the name of the parameter says, you must set the number of pulses that appear on the output of the encoder in one revolution. Default value of this parameter is set to 10000.

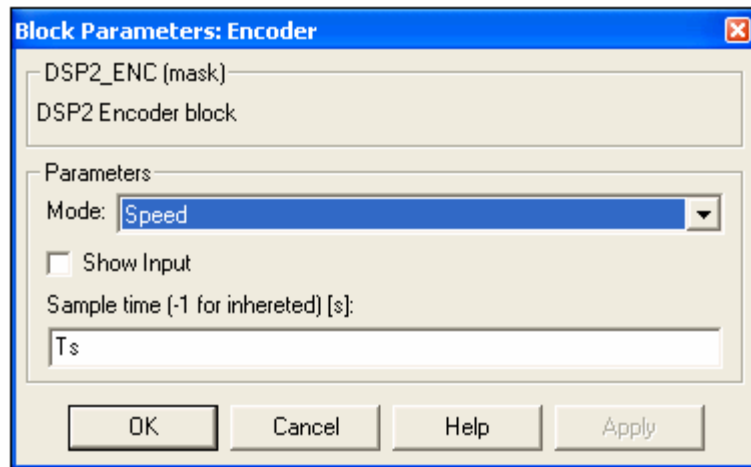


Figure 35: GUI of DSP-2 Encoder block

Parameters: Mode

Speed: if "speed" is selected then speed in rad/s (radian/second) of incremental encoder appears on the output of this block

Position: if "Position" is select then the position of incremental encoder (in radians) appears on the output of block

Gain

With parameter *Gain* we can scale speed or position to the desired units.

Sample time

Sample time must be an integer multiply of the base sample time (*Fixed step size* parameter). If the sample time is set to -1, then the sample time is inherited from the driving block. Default value is '-1'.

Modulator

Description: Block *Modulator* transforms the desired voltage vector, which is given in a-b system of coordinates, to the relative time length of transistors conduction. Desired voltage vector is given with two inputs, while calculated switching times appear on the output of the block. This block is used in combination with DSP-2 PWM block. You can connect outputs of DSP-2 Modulator with T1, T2 and T3 inputs of the DSP-2 PWM block.

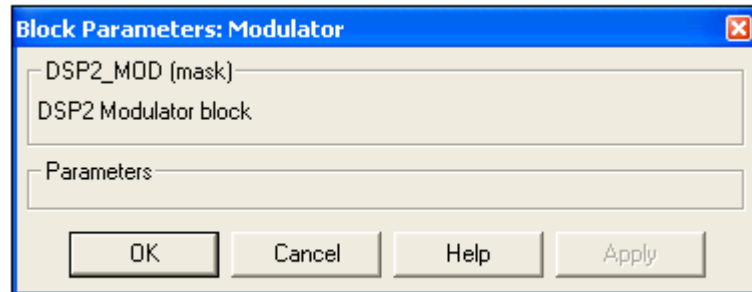
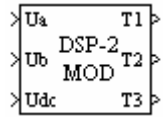
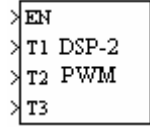


Figure 36: GUI of DSP-2 Modulator block

Parameters: None.

PWM

Description: Block DSP-2 PWM (Figure 37) has only one parameter *Enable level*. When the value of the input signal, which is connected to input EN, is greater than the mentioned parameter, the pulse width modulator, which is located in FPGA, starts working. In each ISR the values of T1, T2 and T3 inputs of this block are transferred to registers W_MOD_T1, W_MOD_T2, and W_MOD_T3 (look at DSP-2 User's guide) of PWM.



Period of the PWM is equal to the half of the interrupt period (*Fixed Step Size* on the *Solver page*)!!! Input ranges of T1, T2 and T3 inputs are from 0 to T_{iMAX} , where T_{iMAX} can be calculated from the following equation:

$$T_{i(MAX)} = \frac{\text{Interrupt period [s]}}{2 \cdot 66.6 \cdot 10^{-9} \text{ [s]}} \quad i = 1, 2, 3$$

When the output of the PWM is set to “1”, the upper transistor is active, and at the value “0” the lower transistor is active. Immediate transition between the conduction of the upper and the lower IGBT transistor is not allowed. Transistors must be in inactive state for a while. Duration of inactive state or ‘dead time’ can be set with the parameter *Modulator dead time* in *Options page* (2) (Figure 17). This option appears only, if the DSP-2 target is selected. Default value of parameter *Modulator dead time* is set to 40 units (1 unit is 66.6ns).

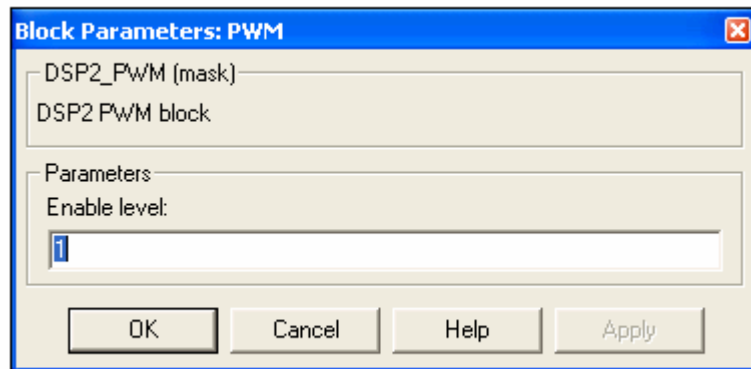


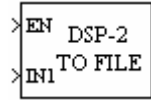
Figure 37: GUI of DSP-2 PWM block

Parameters: Enable level

Parameter must be a positive scalar value. When the value of input signal, which is connected to the input EN, is greater than parameter *Enable level*, then PWM starts working.

To File

Description: Figure 38 shows the GUI of DSP-2 To File block. This is an example of dynamic GUI, because content of GUI is changing dynamically depending on parameter *Num of input vars*.



When the signal, which is connected to input EN, becomes greater than the parameter *Enable level*, all block input signals (except of input EN) are stored to temporary buffer on DSP-2 controller. The size of this buffer can be changed with parameter *Buffer size* on *RTW Options page (2)* (Figure 17). Because DSP-2 controller memory size is limited, it is recommended, that this parameter does not exceed 20000. Input signals share out buffer size in equal parts. That means, if you set *Buffer size*=20000 and *Num of input vars*=4, then each input signal occupies 5000 locations in the buffer.

When the buffer is full, the transfer of buffer data from DSP to PC is performed (DSP Terminal program must run on PC!). Buffer data is store in file with the name that is enter in parameter *File name*.

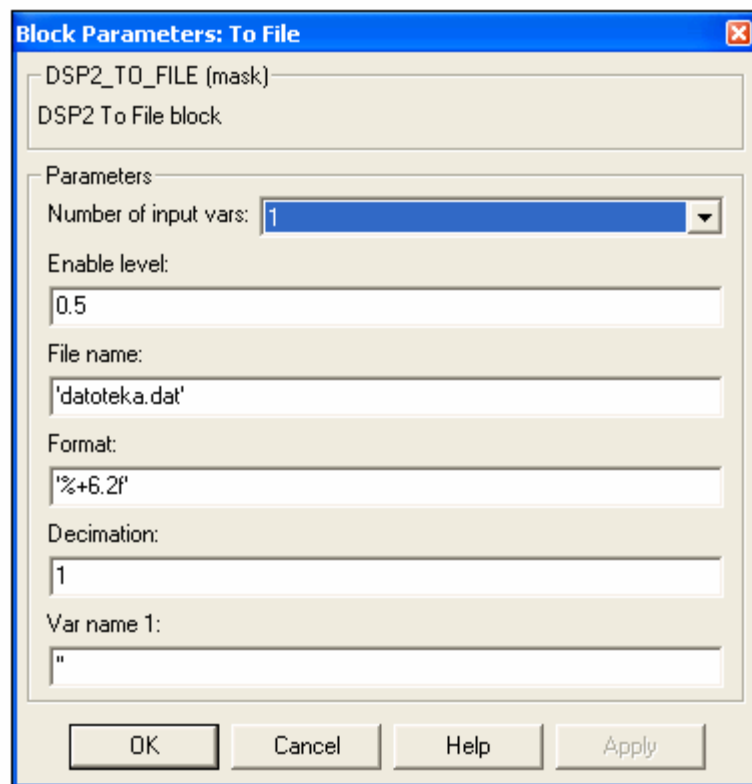


Figure 38: GUI of DSP-2 To File block

Parameters: Number of input vars

Number of block inputs.

Enable level

When the signal, which is connected to the input EN, becomes greater than the parameter *Enable level*, all block input signals (except of input EN) are temporary stored to the buffer.

File name

The name of the file that holds the block input data.

Format

Block input signals are stored to the buffer and then to the file in the »float« format, therefore parameter *Format* must be of type %X.Yf, where X is the number of all, and Y number of decimal places.

Decimation

The *Decimation* parameter is the decimation factor. It can be set to any positive integer *d*, and allows you to write data to buffer in every *d*th interrupt. The default decimation is set to '1'. - writes data to buffer in every ISR.

Var name

The name of input signal

To Terminal

Description: Figure 39 shows GUI of the DSP-2 To Terminal block, which enables sending of the block input data to the serial bus of DSP-2 controller. If DSP Terminal is running on PC, then you can monitor block input data in DSP Terminal *Text* or *Visual* page. In GUI you can set the following parameters: *Target page of DSP Terminal*, *Variable name*, *Format* and *Unit*. All parameters except *Target page of DSP Terminal* must be entered within single quotes.

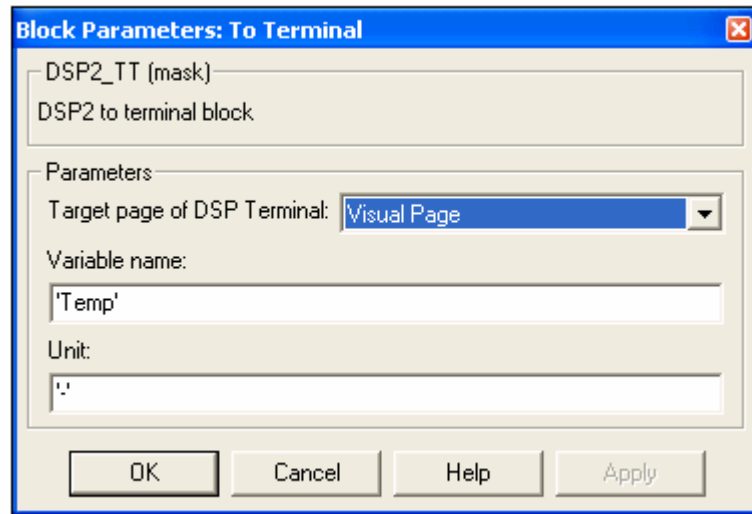
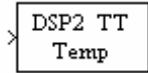


Figure 39: GUI of DSP-2 To Terminal block

Parameters: Target page of DSP Terminal

Text Page: If this option is selected the value of the block input signal appears in Text page of DSP Terminal.

Visual Page: If this option is selected the value of the block input signal will appears in *Visual page* of DSP Terminal.

Variable name

The name of the variable that holds the input data.

Format

Block input signal is stored to variable in »float« format, therefore the parameter *Format* must be of type %X.Yf, where X is the number of all, and Y number of decimal places.

Unit

Unit name appears after the variable value

Example: Let us assume, that we set the block parameters to the following values:

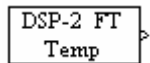
- *Target page of DSP Terminal Variable*='Text Page'
- *name*= ' Current',
- *Format* = %3.1f
- *Unit* ='A'

If in a moment of observation the value of the block input signal is equal to 50, then on the *Text page* of the DSP Terminal appears the next line:

Current=50.0 A

From Terminal

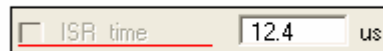
Description:



DSP-2 From Terminal (Figure 40) is source block. In combination with DSP Terminal it enables online changing of the variable values. Block has four parameters: *Variable name*, *Initial value*, *Type* and *Unit*.

Block creates new variable with the name equal to *Variable name* parameter. After the code generation and downloading process, in the upper part of DSP Terminal *Visual page*, for each DSP-2 From Terminal block in the Simulink model, the following objects appear (figure below):

- *Option box*: option is disabled
- *Text box* – the same name as we assigned in the parameter *Variable name*, in GUI of DSP-2 From Terminal block, is displayed.
- *Edit box* – the value of the variable at the time of observation is displayed.
- *Text box*- Unit - the same name as we assign in the parameter *Unit*, in GUI of DSP-2 From Terminal block is displayed.



Now we can change the variable value, by entering different value in *Edit box*.

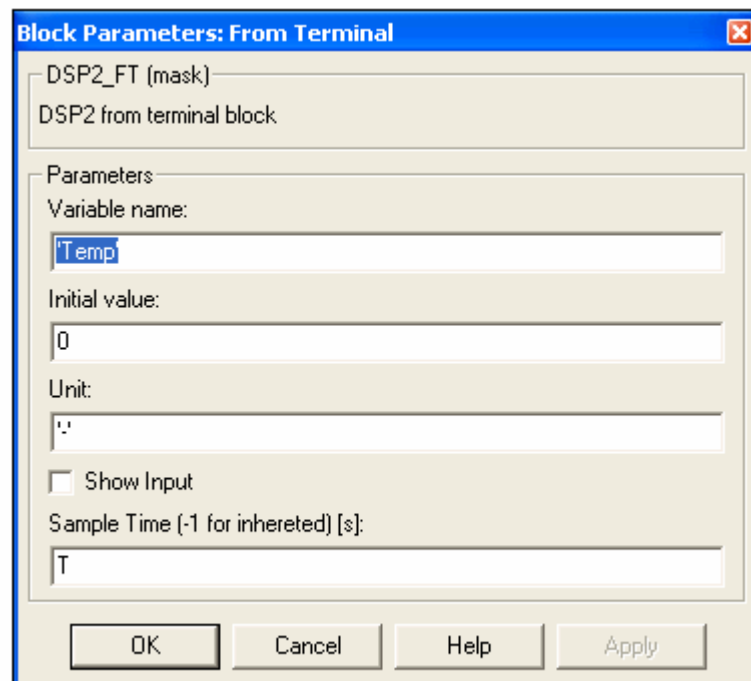


Figure 40: GUI of DSP-2 From Terminal block

Parameters: Variable name

String of alphanumeric characters. Parameter must be entered in single quotes.

Type

This option is not implemented yet.

Initial value

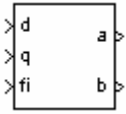
Scalar real number. The value, to which DSP initializes the variable in the initialization stage of DSP-2 controller.

Unit

String of alphanumeric characters. Parameter must be entered in single quotes.

Transformations

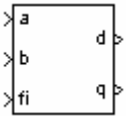
Description: DSP-2 library contains blocks for the following transformations:



- transformation from ab to dq system of coordinates:

$$d = \cos(fi) \cdot a + \sin(fi) \cdot b$$

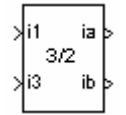
$$q = -\sin(fi) \cdot a + \cos(fi) \cdot b$$



- transformation from dq to ab system of coordinates:

$$a = \cos(fi) \cdot d - \sin(fi) \cdot q$$

$$b = \sin(fi) \cdot d + \cos(fi) \cdot q$$



- transformation from 3 phase to 2 phase system of coordinates:

$$i_a = i_1$$

$$i_b = \frac{1}{\sqrt{3}}(2 \cdot i_3 - i_1)$$

above transformation equation is valid, if: $i_1 + i_2 + i_3 = 0$



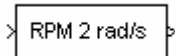
- Conversion from Degrees to Radians

$$R = \frac{\pi}{180} \cdot D$$



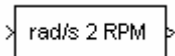
- Conversion from Radians to Degrees

$$D = \frac{180}{\pi} \cdot R$$



- Conversion from RPMS to rad/s

$$rad / s = \frac{\pi}{30} \cdot RPM$$



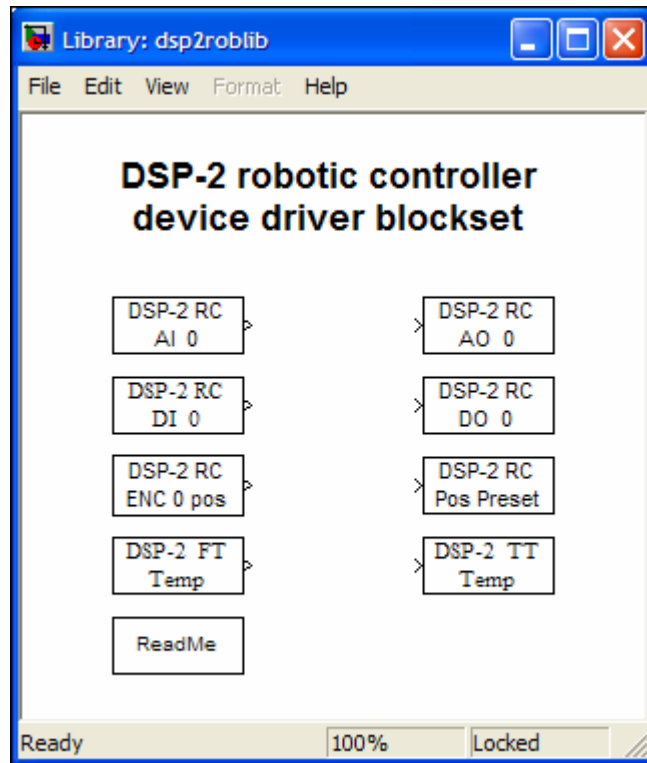
- Conversion from rad/s to RPMS

$$RPM = \frac{30}{\pi} \cdot (rad / s)$$

4.2. DSP-2 robotic controller device driver blockset

DSP-2 robotic controller blocks can be used in Simulink model only, if DSP-2 robotic controller is attached to the DSP-2 controller!

In the following subsections individual DSP-2 robotic controller blocks are explained.



DSP-2 robotic controller blockset contains the following blocks:

- Analog Input
- Analog Output
- Digital Input
- Digital Output
- Encoder
- Encoder Position Preset
- From Terminal
- To Terminal

Figure 41: DSP-2 robotic controller blockset

NOTE: When you are developing code for DSP-2 robotic controller, the following DSP-2 device driver blocks are not allowed in Sumulink model:

- DSP-2 Analog Input/Output
- DSP-2 Digital Input/Output
- DSP-2 CAN Read/Write
- DSP-2 PWM
- DSP-2 PWM2
- DSP-2 Encoder

Analog Input

Description: DSP-2 robotic controller has four 12 bits analog to digital converters (ADC) with the input range from -10 V to 10 V. An output of *Analog Input* block has ranges from -10 V to 10 V.

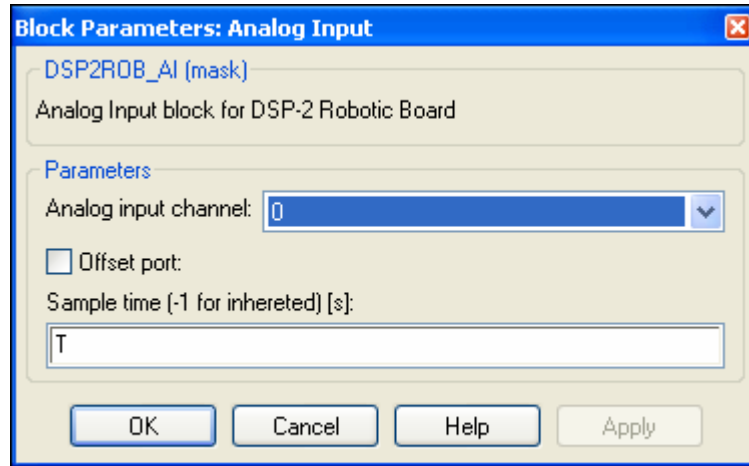
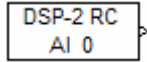


Figure 42: GUI of DSP-2 robotic controller Analog Input block

Parameters: Analog input channel

Analog input channel of DSP-2 robotic controller, from which DSP-2 controller reads data.

Offset port

If this option is selected, block input port appears. Value connected to this input port is subtracted from ADC data.

Sample time

Sample time must be an integer multiple of the base sample time (*Fixed sample time* parameter).

Analog Output

Description: DSP-2 robotic controller has four 12 bit digital to analog converters (DAC), with the output ranges from -10 V to 10 V. With parameter *Analog Output* (Figure 43) can be selected, on which analog channel of DSP-2 robotic controller, block input data will be send. If the block input signal is greater than 10, or smaller than -10, the input signal is limited to 10 or -10 respectively.

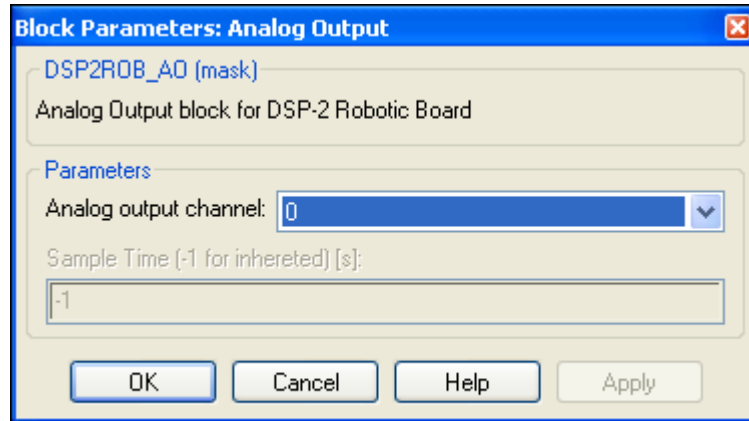
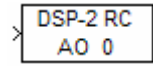


Figure 43: GUI of DSP-2 robotic controller Analog Output block

Parameters: Analog output channel

Analog output channel of the DSP-2 robotic controller , to which DSP sends block input data.

Sample Time

Sample time must be an integer multiply of the base sample time (*Fixed step size* parameter). If sample time is set to -1, then sample time is inherited from driving block. Default value is '-1'.

Digital Input

Description: DSP-2 robotic controller has 16 optically isolated logical inputs. Which input to use, can be selected with the parameter *Digital Input* (Figure 44). On the output of this block, signal with the logical value '0' or '1' appears.

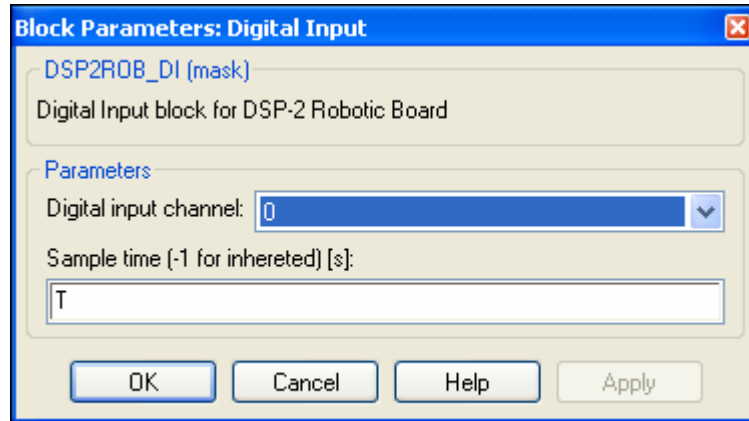
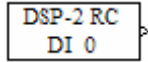


Figure 44: GUI of DSP-2 robotic controller *Digital Input* block

Parameters: Digital input channel

Digital input channel of the DSP-2 robotic controller , from which DSP gets data.

Sample time

Sample time must be an integer multiple of the base sample time (*Fixed sample time* parameter).

Digital Output

Description: DSP-2 robotic controller has 8 optically isolated logic output. In the GUI of this block (Figure 45) the following parameters can be set: *Digital output*, *Threshold* and *Operation*. The value of the output signal is depending of height of the input signal, operation and threshold parameters.

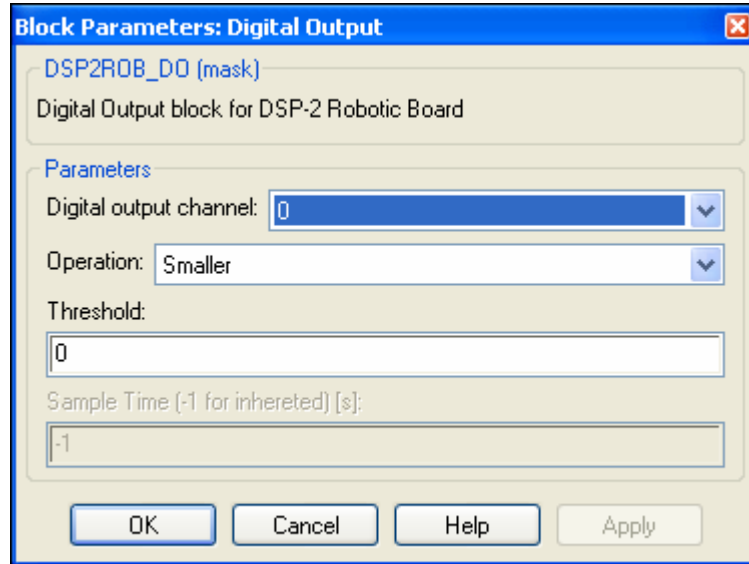
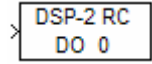


Figure 45: GUI of DSP-2 robotic controller Digital Output block

Parameters: Digital output channel

Select to which digital output channel of DSP-2 robotic controller, DSP will sends data

Operation

Greater : If the input signal is greater than the threshold, then the output is set to '1' else to '0'

Smaller: If the input signal is smaller than the threshold, then the output is set to '1' else to '0'

Threshold

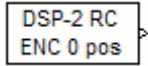
Threshold must be a scalar real number. The meaning of this parameter is explained in *Operation* parameter.

Sample time

Sample time must be an integer multiply of the base sample time (*Fixed step size* parameter). If the sample time is set to -1, then the sample time is inherited from the driving block. Default value is -1.

Encoder

Description: Block *DSP-2 Encoder* enables sampling of the position and speed of incremental encoders that are connected to the DSP-2 robotic controller. In block GUI (Figure 46) you can set two parameters: *Encoder* and *Mode*. The meaning of these parameters is explained in Parameters section below.



To ensure the correct operation of incremental encoders, *Num of lines per revolution* and *Hardware Multiplication* parameters, for each connected encoder must have been defined on *DSP-2 robotic controller options (i)* ($i=1..4$) (Figure 20, Figure 21, Figure 22, Figure 23). Note, that these parameters appear only, if the DSP-2 controller target is selected.

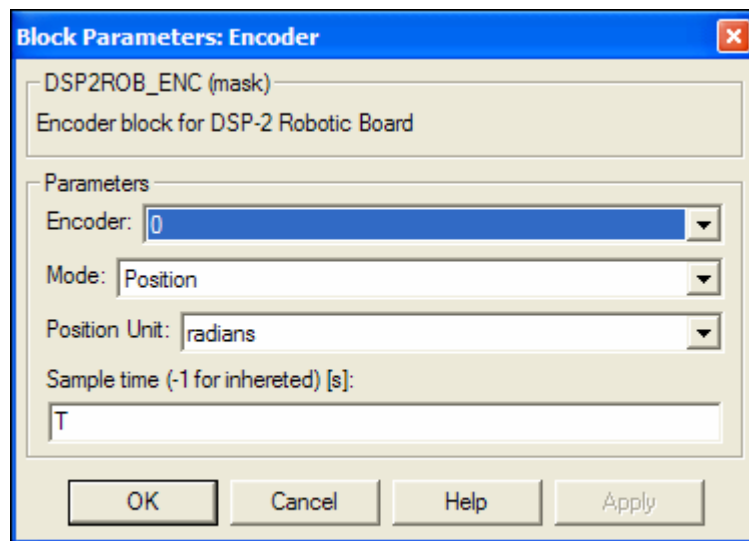


Figure 46: GUI of DSP-2 robotic controller Encoder block

Parameters: Encoder

Encoder of DSP-2 robotic controller , from which DSP gets data

Mode

Speed: if "speed" is selected then speed in rad/s (radian/second) of incremental encoder appears on the output of this block

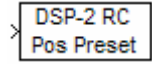
Position: if "Position" is select then the position of incremental encoder in appears on the output of the block. With the parameter **Position Unit** the user can define position in radians or in pulses.

Sample time

Sample time must be an integer multiply of the base sample time (*Fixed step size* parameter). If the sample time is set to -1, then the sample time is inherited from the driving block. Default value is '-1'.

Encoder Position Preset

Description: Block *Encoder Preset* enables encoder i ($i=0..3$) position preset. In block GUI (Figure 46) you can set two parameters: *Encoder* and *Consider reference signal*. The meaning of these parameters is explained in Parameters section below.



To ensure the correct operation of incremental encoders, *Encoder i Position Preset Value* ($i=0..3$) and/or *Invert reference signal* parameters, for each connected encoder must have been defined on *DSP-2 robotic controller options* (Figure 20, Figure 21, Figure 22, Figure 23). Note, that these parameters appear only, if the DSP-2 controller target is selected.

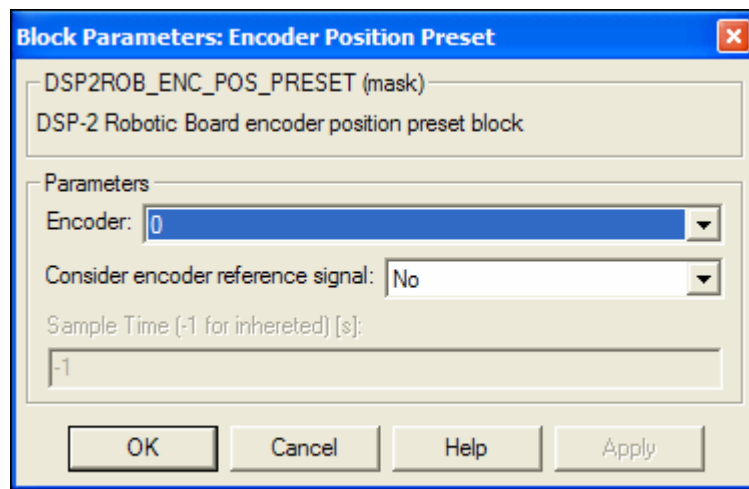


Figure 47: GUI of DSP-2 robotic controller Encoder Preset block

Parameters: Encoder

Encoder of DSP-2 robotic controller

Consider reference signal

No: If this option is selected, encoder position will be preset, on the positive slope of the block input signal, to the value that is defined in parameter *Encoder i Position Preset Value* ($i=1..3$). Note: Use this option only, if your encoder does NOT contains reference signal.

Yes: If this option is selected, encoder position preset mode will be started on the positive slope of the block input signal. Afterwards, encoder position will be preset to the value defined in parameter *Encoder i Position Preset Value* ($i=1..3$), when the encoder reference signal will appear (disappear).

Note: If you use this option, and position preset occur immediately after positive slope of block input signal, turn *Invert reference signal*

parameter on *DSP-2 robotic controller options* (i=1..3) page to the opposite value.

Sample time

Sample time must be an integer multiply of the base sample time (*Fixed step size* parameter). If the sample time is set to -1, then the sample time is inherited from the driving block. Default value is '-1'.

4.3. DSP-2 Plants

In addition to the DSP-2 learning module, Institute of Robotics provides DSP-2 plants which can be easily connected to DSP-2 learning module. These plants are Buck converter, DC motor (Figure 49), ADDA interface card (Figure 48), etc. Documentation of these plants, which include description of Simulink blocks and description of demonstration examples, could be found on the following internet address:

<http://www.ro.feri.uni-mb.si/projekti/dsp2/documentation.htm>

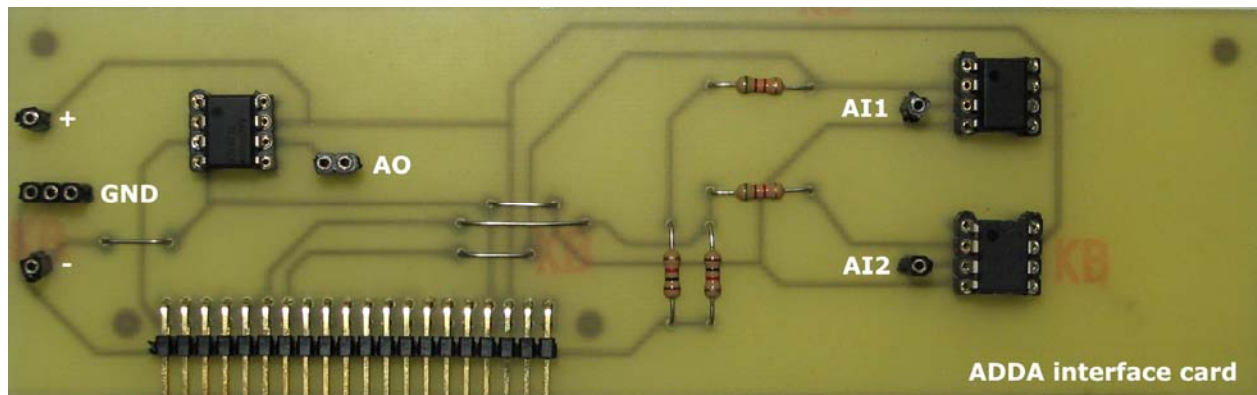


Figure 48: ADDA interface card

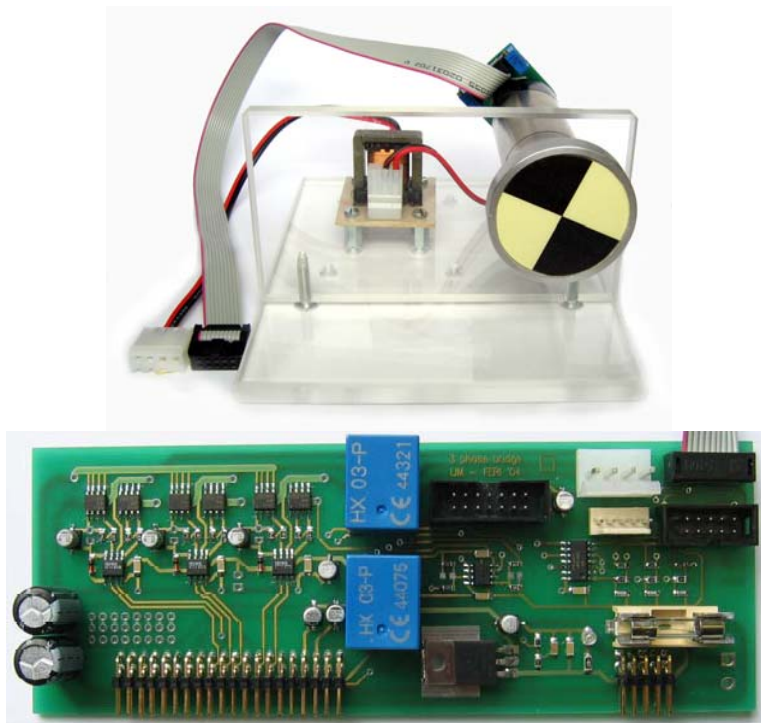


Figure 49: DC motor with 3 phase H bridge

5. Blocks that are not supported with *Embedded C format*

Code for '*Embedded C format*' does not support the following blocks:

- Continuous
 - o No blocks are supported!
- Discrete
 - o First-Order Hold
- Functions and tables
 - o MATLAB Fcn
 - o Following S-function: M-files or Fortran S-function, and noninlined C-MEX S-functions.
- Math
 - o Algebraic Constraint
 - o Matrix Gain
- Nonlinear
 - o Rate Limiter
- Sinks
 - o XY Graph
 - o Display
- Sources
 - o Clock
 - o Pulse Generator
 - o Ramp
 - o Repeating Sequence
 - o Signal Generator